# Free-Wake Computation of Helicopter Rotor Flowfield for General Flight Regimes

Final Report

K. Ramachandran

J. S. Steinhoff

W. Yonghu

30 Sept. 1992

U. S. Army Research Office

DAAL03 - 89 - C - 0027

Flow Analysis Inc.
256, 93 Street
Brooklyn
NY 11209

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

# Free-Wake Computation of Helicopter Rotor Flowfield for General Flight Regimes

Final Report

K. Ramachandran

J. S. Steinhoff

W. Yonghu

30 Sept. 1992

U. S. Army Research Office

DAAL03 - 89 - C - 0027

Flow Analysis Inc.
256, 93 Street
Brooklyn
NY 11209

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

**1. AGENCY USE ONLY** *(Leave blank)*

**2. REPORT DATE**
March 30 1993

**3. REPORT TYPE AND DATES COVERED**
Final 1 Oct 89 - 30 Sep 92

**4. TITLE AND SUBTITLE**
Free-Wake Computation of Helicopter Rotor Flowfield for General Flight Regime

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Krishna Ramachandran
John Steinhoff
Wenren Yonghu

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Flow Analysis Inc.,
256 93 Street,
Brooklyn ,
NY 11209

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
ARO 25623.4-EG-S

**11. SUPPLEMENTARY NOTES**
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

A CFD based computer code, HELIX II has been developed for computing helicopter rotor flowfield in hover and in forward flight. In order to predict the vibratory airloads in forward flight and the performance in hover the effect of shed wake has to be accurately accounted for. This has been accomplished by an unique vorticity embedding procedure. The wake is convected in a Lagrangian sense and does not suffer from numerical dissipation. The effect of this convected wake is included in the Eulerian using Clebsch representation. The unsteady full potential equation is solved in the Eulerian grid with the embedded vortical velocity accounting for the influence of wake. The wake evolves as a part of the solution. Dynamic grid modifications are performed at each time step to account for blade motion and deformation. These modifications are confined to regions near the blade using suitable blending functions.

In addition a vorticity confinement method is developed which involves adding a term to the momentum conservation equation of fluid dynamics. This modification is very useful in the numerical solution of flow problems involving vortical regions. The Euler equations solved with this extra term can convect vortices without Numerical dissipation.

**14. SUBJECT TERMS**
unsteady airloads, Lagrangian wake convection, Eulerian grid,
Vorticity Embedding, Clebsch representation
Vorticity confinement, Euler Equation solution without dissipation

**15. NUMBER OF PAGES**
42 + APPENDIX 30 72

**16. PRICE CODE**

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

# Table of Contents

# List of Figures

# 1. Statement of the problem

The accurate computation of rotor flows requires the proper treatment of strong, concentrated vortex sheets that are produced by rotor blades and convect near the blades. For modern rotors, it also requires the proper treatment of compressibility effects, including shocks, which can occur near the blades. In addition, it requires the proper treatment of blade dynamics. Different methods have existed for some time for separately treating each of these effects. No method, however, has been able to treat them in combination and provide the total analysis that rotorcraft require. The availability of a comprehensive aerodynamic code that can meet the above challenges would be of great benefit to the helicopter industry.(1,2) It would greatly reduce many uncertainties in helicopter design and also reduce much of the dependence on expensive and time consuming wind tunnel testing. This would allow more efficient designs to be developed. To be used, however, such a code should be validated against experiment, so that engineers have confidence in its results. It must also be reliable and robust enough to be usable by engineers in the design environment, without requiring empirical inputs for each different case.

This report concerns the development of such an analysis tool, using a new CFD methodology termed "Vortex Embedding" which has been developed and validated over the last several years, for the hovering rotor problem. The new code, HELIX-II is the forward flight version of the hover code and the original algorithm has been extended to study the forward flying helicopter rotor. In this report three major issues concerning this problem have been addressed:

1. Detailed unsteady compressible free- wake computations with strong vortical effects but without actual impingement with the rotor blade.

2. Incorporating the blade motion terms such as cyclic pitch variations and flapping. This scheme is general and can be used to include aeroelastic effects.

3. A separate CFD scheme has been developed to demonstrate vortex impingement on the rotor blade.

## 2. Background

Euler or Navier-Stokes equation solution methods using surface fitted, fixed (Eulerian) grids have been shown to give accurate results for compressible flows, even in the presence of shocks. However, when concentrated vortices are present, these fixed grid, (or Eulerian) methods result in large amount of unphysical numerical diffusion, unless high order schemes are used, together with a dense computational grid. In order to prevent this diffusion:

- First, a higher order discretization (fifth) together with a fine , regular grid in the entire space where the vortex travels has been used to solve the general problem of a vortex impinging on an airfoil.[3] This, of course, requires a large amount of computing, although fairly efficient implicit solvers can be used since the grid is regular.

- Second, a more efficient utilization of grid points can be made with an adaptive scheme where grid points are clustered in regions of high vorticity, and a Navier-Stokes solver used.[4] Here, a relatively a large number of grid points must still be allocated to the vortical regions to prevent diffusion. Since, for many problems, the vortices can propagate over relatively large distances before impinging on a surface these methods still require large computing resources. Further, for general vortical flows, unstructured grids are required to achieve significant clustering. This results in complex logic and book-keeping. Also, without regular grids, conventional, efficient implicit solution methods cannot generally be used. The explicit solution methods which then must be used are much less efficient and result in much longer computing times, for realistic 3-D problems.

These two strategies are the only ones, which have been used for general compressible strong vortex interactions in aerodynamics - where the internal structure of the vortex is involved and must be solved for - and which avoid significant numerical diffusion.

Potential Flow methods also use fixed, or Eulerian grids, are fully compressible and can capture shocks.[5] Contact discontinuities, or vortex sheets, however, are normally treated as potential discontinuities and do not diffuse. In normal treatment, they are fixed on grid planes and do not follow the flow. Compressible Potential Flow solutions conserve mass throughout the field as do Euler/Navier-Stokes methods. They also conserve momentum everywhere that there are no vortex sheets (unless very strong shocks are present). Because potential discontinuities are constrained to lie on grid surfaces, however, conventional Potential Flow solutions do not conserve momentum through vortex sheets and cannot be used where these vortices convect close to other surfaces and cause large effects.

Because of the diffusion or constraint problems associated with the treatment of vortex sheets, the above methods, by themselves, are not suitable for treating rotor problems, where the locations and strengths of these sheets must be accurately computed.

Currently, the most successful CFD method for the treatment of rotor-wake problems involves the use of vorticity embedding(VE). This method is unique among CFD methods in that it preserves wake circulation without requiring dense grids. This is because the circulation is not carried by the grid, but rather by a sheet of convecting wake markers, whose circulation is then impressed on the adjacent grid points as a local vortical velocity distribution. This approach, used by the HELIX-I(6,7,8) code has been successfully applied for the prediction of hover wakes and performance.

In Fig. 1 and Fig. 2 the computed load distribution and the performance using vortex embedding are compared with experimental data.

Forward flight is much more computationally intensive than hover for several reasons. For hover any number of blades can be computed(at no extra cost) simply by appropriate specification of boundary and wake periodicity. This is not possible in forward flight. Furthermore, in hover the end result is a single steady flow solution (in rotating coordinates) with one wake geometry. In forward flight, there is no steady flow solution or a single wake geometry. The wake system is different at each time step and the solution must be constantly reconverged to accommodate this changing wake. This necessitated several modifications to the original approach (HELIX-I) in order to improve the code computational efficiency and the capability to include unsteady wakes. These have been implemented in the code HELIX-II.

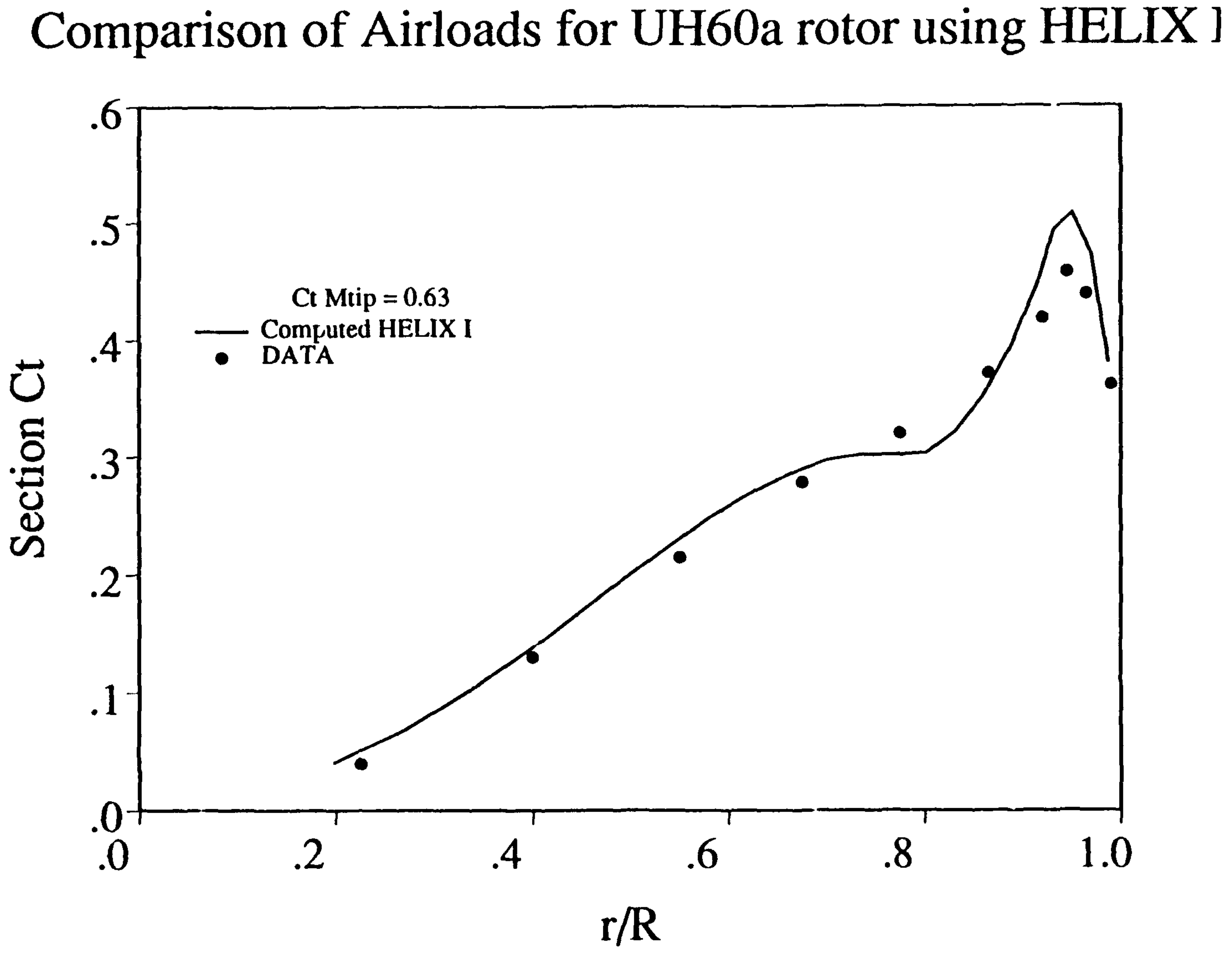


Figure 1: Comparison of spanwise loading in hover for a 4 bladed rotor, Mtip = 0.63

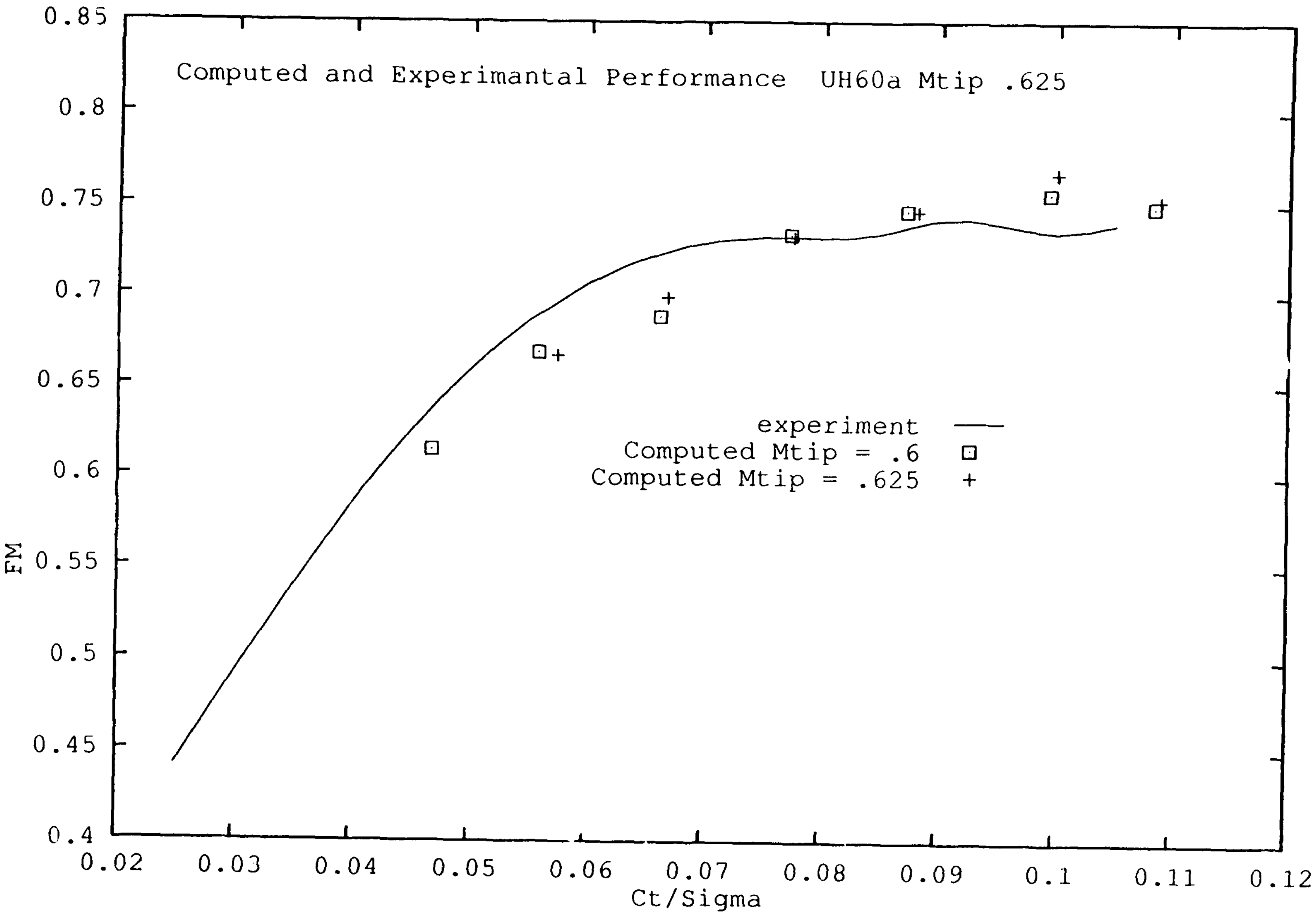


Figure 2: Comparison of performance in hover for a 4 bladed rotor

## 3. Vortex Embedding Method for Advancing Rotor Blades

Recently, several new schemes have been developed(9,10) wherein an inner CFD based method is coupled to either a comprehensive rotor code or a free-wake method based on a lifting surface type Lagrangian method.

In our method, a Lagrangian wake convection scheme is incorporated in a potential flow based method and thus no external coupling is required to include the wake effects. At low advance ratio forward flights the wake effects are important as in the case of hover and need to be modeled properly. But unlike hover the solution is no longer axially symmetric and the wake is different for each time step and the solution must be constantly reconverged at each time step to accommodate this changing wake. With this in mind the new algorithm was developed that is computationally efficient and as accurate as the original algorithm.

Some important features of HELIX-II are :

1. The unsteady full potential equation is solved by a semi-implicit method based on approximate factorization.

The solver has been modified to include multiple blades. Thus the code computes entire 360 degrees.

At each time level Newton type sub-iterations are performed to achieve time accuracy and to obtain the correct wake geometry.

A local time linearization provides a good initial guess for the Newton iteration.

An improved wake convection logic provides a smooth transition from the potential jump representation to the Lagrangian wake.

2. The wake is represented by a set of markers distributed along the spanwise and azimuthal directions. Using an initially specified marker

strength (from a hover calculation) and the undistorted wake geometry a vortical velocity is computed using Clebsch's variables,

$$\vec{q^v} = \Gamma\vec{\nabla}\lambda \qquad \text{----- ( 1 )}$$

Since a major portion of the computing time is spent on determining this vortical velocity a new search and spreading algorithm which is faster than the original approach has been developed using computational coordinates instead of physical co-ordinates. This speed up is important for computational efficiency because in forward flight it is required to update the wake and the vortical velocity at every time step. The vortical velocity accounts for the wake effects in the solution to the unsteady mass conservation equation in the Eulerian grid.

3. Due to prescribed blade motion, flapping and structural deformations, the blade attitude has to be modified at each time step. These effects are easily included by modifying tthe grid in the vicinity of the blade. Simple blending functions are used to keep the outer regions unchanged.

A schematic of the HELIX-II is shown in Figure 3. It can be seen from this that each time step of free-wake calculation involves several modular computations and by lagging them by one time step these computations can be performed independently at the same time. The steps are :

1. The time step is initialized. An undistorted wake is obtained using the advance ratio and rotational speed. This undistorted wake comprises of nodes distributed in the azimuthal and radial directions. The radial nodes are located at cell centers and are distributed at constant intervals in the azimuthal direction and this interval is determined from the time step. A vortical velocity is obtained at the grid nodes using this wake and a detailed description of this procedure is given in the following section.

2. Blade motion terms like flapping and cyclic pitch are determined either from a coupled dynamic code or externally specified inputs. They are included as grid changes.

3. The unsteady mass conservation equation is solved for the potential using a finite volume scheme, with vortical velocity representing the wake. A

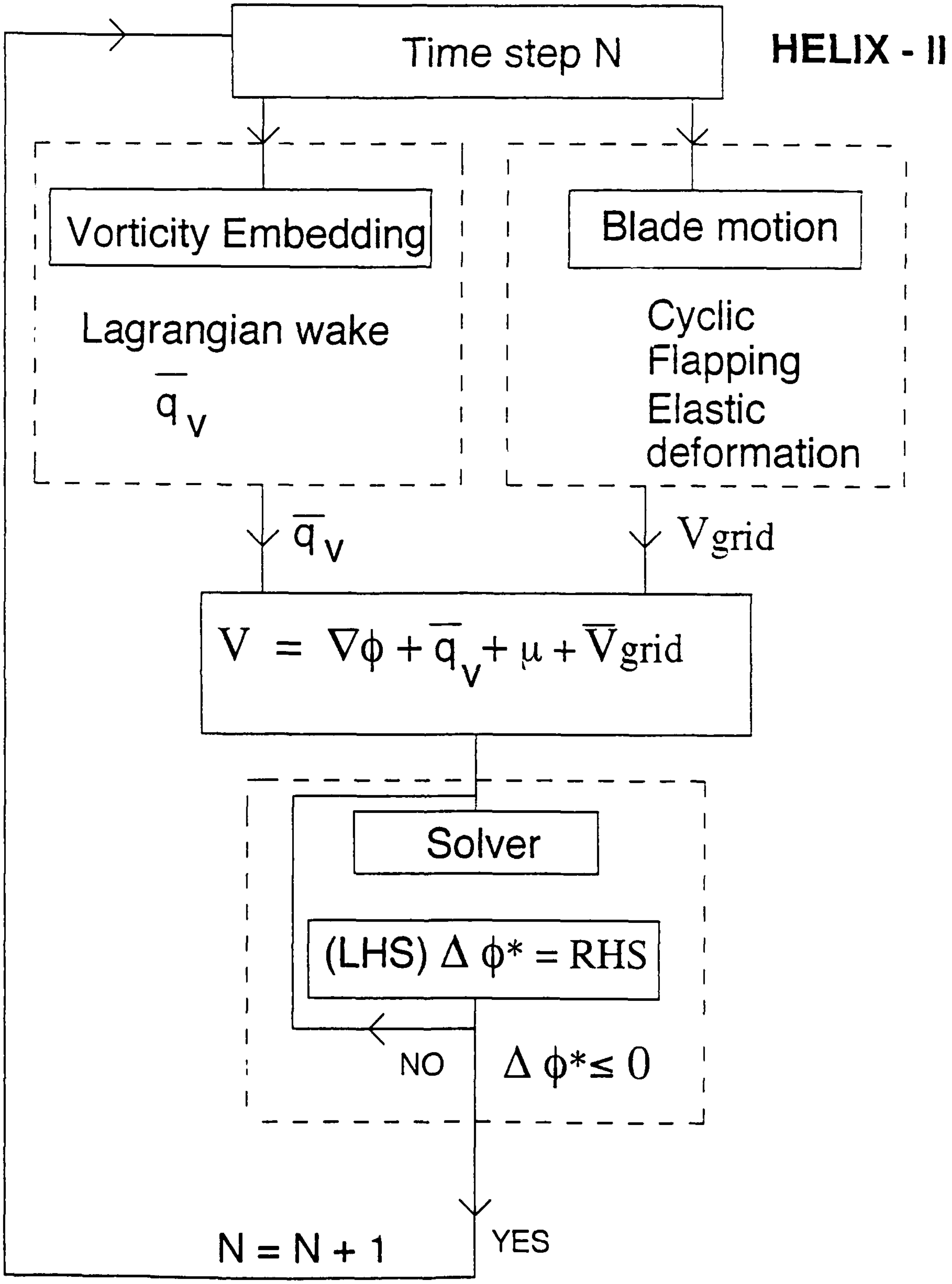


**Figure 3: Schematic of HELIX-II solution procedure**

time linearization and Newton sub-iterations are done while solving the full potential equation. This procedure is described in section 3.3.

4. From the potential, its gradient and the total velocity are obtained and used for wake deformation.

5. Time step is advanced. A new node is added to the wake, corresponding to the trailing edge location of the blade at the current time step. Wake nodes are ddistorted to follow the local flow.

6. A new $\bar{q}^{V}$ is calculated. Return to step 2. This process is continued for several blade revolutions until the blade is trimmed and loading converges. That is, when the azimuthal loading repeats between the revolutions the computation is stopped.

Each step is separately explained in the following sections. Since the approach is modular each module can separately be upgraded or modified with an enhancement without affecting other modules. The application of HELIX-II for a two bladed rotor blade is shown in Chapter 4.

## 3.1 Wake Convection

As mentioned earlier, the wake is represented by a set of markers distributed along the spanwise and azimuthal directions. This initial wake is undistorted and computed using rotational and forward velocities. Each marker node is assigned an initial circulation. This circulation, if obtained from hover calculation is constant in the azimuthal direction and varies only in the spanwise direction. It is updated every time step and this variation accounts for the shed vorticity. This Lagrangian wake nodes are identified in the Eulerian rotor fixed grid. This step involves a search procedure that first identifies the cell.

Once the cells are identified each Lagrangian wake node is assigned a value in computational coordinate system.

$$\vec{X}_{L,LV} = I_{L,LV} + \vec{A}_{L,LV}$$

where $I_{L,LV}$ refers to the Eulerian grid cell the marker node (L,LV) is located and $\vec{A}_{L,LV}$ is the distance from each face of the cell. With $\vec{X}_{L,LV}$ and $\Gamma_{L,LV}$ describing the Lagrangian wake, they are used to determine the vortical velocity in the Eulerian grid. This procedure is described in the next section.

The wake convection procedure for a new time step involves the following steps:

1. The inertial coordinates of the blade fixed grid is obtained corresponding to the new time step.

2. A set of new marker nodes is added. They correspond to the blade trailing edge location at each spanwise station of the inertial grid.

3. All marker nodes are displaced to follow the local flow. That is, given the inertial grid, using a search procedure the location of the Lagrangian nodes are determined in the Eulerian grid. Then the grid velocities (3 components) are interpolated to the wake nodes using a trilinear interpolation. The new wake coordinates are given by

$$\vec{X}_{L,LV,t+\Delta t} = \vec{X}_{L,LV,t} + \vec{V}_{L,LV} . \Delta t \qquad \text{----- ( 2 )}$$

where $\vec{V}_{L,LV}$ is the interpolated velocity.

During this step, in addition to obtaining the inertial coordinates of the wake, the cell identifying procedure provides the marker computational coordinates. They are used for vortical velocity computation. These integrated marker nodes define the new wake for the present time step. The wake coordinates along with blade loading determine the convergence of the solution. In this module, the search algorithm requires the maximum amount of computing time. This process involves intensive vector and geometric computations. The present algorithm is very fast and general. Also, exhaustive search is performed only once in the beginning and subsequent searches use information from the earlier computation.

In HELIX - I, the time step for Lagrangian integration was chosen based on the local grid spacing in the azimuthal direction. This flexibility provided a greater accuracy in the wake geometry in regions near the blade where the grid is dense. In HELIX - II, a uniform azimuthal node distribution is needed for time accurate computations and the original approach is no longer practical. When a large time step ( like 4 degrees ) is chosenthis may result in poor resolution in regions near the blade where the grid is dense and a 4 degree interval may skip the entire blade. Without enough nodes, the resulting $\vec{q}^V$ computation will be inaccurate as the nodes may 'jump' several computational cells in 1 time step. In order to prevent this, a sub interpolation scheme is used which sub divides the computed wake between nodes. The wake with these added sub nodes mimics the local time stepping approach of HELIX - I and is used only for vortical velocity computation. The subdivision can be described as follows:

Let $\vec{X}_{CL,LV}$ be the computed computational coordinate of the wake node ( L, LV ). 'L" is the azimuthal index and LV is the radial index.

Define,

$$N_j = X_{C_{L+1,LV}} - X_{C_{L,LV}}$$

If $N_j$ is greater than 1 , $N_j$ number of 'sub' nodes would be added between nodes L and L + 1. These subnodes will have the grid indices as their X coordinates and the other two coordinates are obtained using weighted interpolation. That is :

$$\vec{X}_C^S(L,LV) = \vec{X}_C(L+1,LV) * F_1/F_3 + \vec{X}_C(L,LV) * F_2/F_3$$

where $F_1$ , $F_2$ and $F_3$ are weighting functions. $\Gamma(L,LV)$ the wake node circulation is also interpolated in a similar manner to the sub nodes. This new wake with atleast 1 sub node in each cell is used for $\vec{q}^V$ calculations.

## 3.2 QV Calculation

Once the wake is obtained for any given time step, the calculation of $\vec{q}_V$ is performed. That is, the circulation of convecting Lagrangian wake nodes are impressed on the adjacent grid points as a local vortical velocity distribution. This step involved a search procedure in the original algorithm that was computationally intensive. In that approach the physical

Lagrangian inertial coordinates were used and an exhaustive search over the entire grid would determine the wake nodes that contributed to the vortical velocity at each grid point. Since for hover, the wake is axially symmetric the final solution involves a single wake geometry, this procedure is still acceptable and simple to implement. The steps involved are:

1. Compute Lagrangian wake $\vec{X}_W$ (L,LV)

2. For each (i,j,k) of Eulerian grid $\vec{X}_E(i, j, k)$ compute the distance $\Delta s$ from an $\vec{X}_W$( wake ) panel.

If ( $\Delta s > a$) set,

$$\vec{q}^{\,V} = 0.$$

where 'a' is a specified smearing distance

3. If $\Delta s < a$, compute,

$$\vec{q}^{\,V} = \Gamma_{ijk} \nabla \lambda_{i,j,k}$$

where $\Gamma$ is interpolated circulation and

$$\lambda = \frac{1}{2} \sin \frac{\pi}{a} \vec{S}_n$$

$\vec{S}_n$ is the normal distance of panel from a given node. The details of this procedure can be found in Ref. 6.

In forward flight, the wake is time dependent and there is no single wake geometry. One trim calculation may involve several azimuthal revolutions and each revolution require as many wake computations as the number of time steps. So the procedure outlined earlier will be prohibitively expensive and impractical. So a revised scheme has been developed which is memory intensive (requires lot more storage) but much faster. In this:

1. The wake convection and the subsequent smearing to compute Clebsch's constants are done simultaneously. That is, in order to convect the wake the grid velocities will have to be interpolated to the marker location. During this process the Lagrangian nodes location are obtained in the index (computational) space. They are saved as $\vec{X}C$ (L, LV).

2. Now obtain,

$$IC\,(L, LV) \quad = \quad INT\,(\vec{X}C\,(L, LV))$$

3. From each IC(L, LV) node define a smearing region in computational space given by

$$\vec{R} \;=\; \vec{I}C \pm D \qquad\text{----- ( 2 )}$$

where 'D' is the number of cells over which the $\vec{q}^{\,V}$ will be spread in the computational space.

4. Obtain Clebsch's variables $\Gamma_{ijk}$ and $\lambda_{ijk}$ at these cell nodes in computational space using Eqn 1. Since $\Gamma$ and $\lambda$ are scalars they are invariant and can be transported to the rotor fixed grid to determine $\vec{q}^{\,v}$.

5. A grid node can get contribution from more than one Lagrangian wake node. Hence these contributions are added and a $\Gamma$ and $\lambda$ are obtained using interpolation like formulae :

$$\Gamma_{ijk} \;=\; \frac{\Sigma\, \Gamma\, W_{ijk}}{\Sigma\, W_{ijk}}$$

$$\overline{\lambda}_{ijk} \;=\; \frac{\Sigma S_n\, W_{ijk}}{\Sigma\, W_{ijk}} \qquad\text{----- ( 3 )}$$

$$\lambda_{ijk} \;=\; \frac{1}{2}\sin\frac{\pi\;\overline{\lambda}_{ijk}}{D\sqrt{2}.\;2}$$

where $W_{ijk}$ is a weighting function given by,

$$W_{ijk} \;=\; \left(1 - \frac{\Delta S^2{}_{ijk}}{a^2}\right)$$

$\Delta S_{ijk}$ is the distance of a wake node from the grid point.

$\vec{S}_N$ is the signed normal distance of a grid node from a wake panel.

$\Gamma$ is the strength of the wake node under consideration. It has to be observed that these functions have the same definitionas the original approach except that they are defined in computational space and not physical space. Finally, $\bar{q}^{v}$ is computed using

$$\vec{q}^{\,v} = \Gamma_{ijk} \nabla \lambda_{ijk} \qquad \text{----- ( 4 )}$$

Here, the gradient is computed using the same box scheme as used for the potential calculation.

## 3.3 Solution to Full Potential Equation with $\bar{q}^{V}$

Once $\bar{q}^{V}$ is obtained for a given time step, the time dependent mass conservation equation is solved with $\bar{q}^{V}$ and grid motion terms. That is, solve for,

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} . ( \rho \vec{v} ) = 0 \qquad \text{----- ( 5 )}$$

where,

$$\overline{V} = \overline{\nabla} \phi + \vec{q}^{\,v} + \vec{V}_{BM} \qquad \text{----- ( 6 )}$$

where $\vec{V}_{BM}$ represents the velocity terms from the blade motion. Salient features are :

- Equation ( 5 ) is discretized using a finite volume scheme.

- At each time step local time linearization provides good initial guess for the Newton sub iterations.

- Circulation convection is solved in regions in the wake before being replaced by vortical velocity.

### 3.3.1 Time Linearization

In this section the time linearization procedure for Newton iterations and for achieving time accuracy is outlined. A detailed description of this procedure can be found in Ref. 11.

If n is the running index in the time direction Eqn.( 5 ) can be represented as

$$f(\phi) = 0 \qquad \text{----- ( 7 )}$$

where $\phi$ is the unknown to be solved for at every grid point in the (n+1) time level. The Newton iteration for solution to Eqn. ( 6 ) is

$$F(\phi*) + \frac{\partial F}{\partial \phi}\Big|_{\phi = \phi*} (\phi - \phi*) = 0$$

where $\varphi*$ is the currently available value of $\phi$ at the (n+1) level. At convergence,

$$\Delta\phi = \phi - \phi* \quad \text{will approach zero.}$$

Equation ( 5 ) can be discretized as :

$$\frac{\rho^{n+1} - \rho^{n}}{\Delta\tau} + \left| \frac{\partial}{\partial\xi}(\rho hU) + \frac{\partial}{\partial\eta}(\rho hv) + \frac{\partial}{\partial\zeta}(\rho hw) \right|^{n+1} = 0 \quad \text{---- ( 7 )}$$

using the expansion for $\rho^{n+1}$ etc.,

$$\frac{\rho(\phi*) - \rho^{n}}{\Delta\tau} + \frac{\partial\rho}{\partial\phi}\Big|_{\phi = \phi^{*}} \Delta\phi + f + \frac{\partial f}{\partial\phi}\Delta\phi^{*}\Big|_{\xi} + g + \frac{\partial g}{\partial\phi}\Delta\phi^{*}\Big|_{\eta}$$

$$+ p + \frac{\partial p}{\partial\phi}\Delta\phi^{*}\Big|_{\zeta} = 0$$

where,

$$f = \rho Uh$$
$$g = \rho Vh$$
$$p = \rho Wh$$

are the flux terms wit<sup></sup>ʰ

$$h = \frac{1}{J}, \quad \text{the Jacobian of transformation.}$$

Using:

$$\left.\frac{\partial\rho}{\partial\phi}\right|_{\phi=\phi^*} = \left[-\frac{\rho}{a^2}\left\{\frac{1}{\Delta\tau} + U\frac{\partial}{\partial\xi} + V\frac{\partial}{\partial\eta} + W\frac{\partial}{\partial\zeta}\right\}\right]_{\phi=\phi^*}$$

and similar expansion for $\frac{\partial f}{\partial\phi}$, $\frac{\partial g}{\partial\phi}$, and $\frac{\partial p}{\partial\phi}$ and rearranging and retaining terms containing $\Delta\phi^*$ on the LHS we get,

$$\left\{\begin{array}{l} I + \Delta\tau\left(U^*\frac{\partial}{\partial\xi} + V^*\frac{\partial}{\partial\eta} + W^*\frac{\partial}{\partial\zeta}\right) - \\ \frac{\Delta\tau^2}{\hat\rho}\left(\frac{\partial}{\partial\xi}(\hat\rho A_1)^*\frac{\partial}{\partial\xi} + \frac{\partial}{\partial\eta}(\hat\rho A_2)^*\frac{\partial}{\partial\eta} + \frac{\partial}{\partial\zeta}(\hat\rho A_3)^*\frac{\partial}{\partial\zeta}\right)\end{array}\right\}\Delta\phi^*$$

$$= \frac{\Delta t\, a^2}{\hat\rho}\left(\hat\rho^* - \hat\rho^n\right) + \frac{a^2\Delta t^2}{\hat\rho}\left(\frac{\partial}{\partial\xi}(\hat\rho U) + \frac{\partial}{\partial\eta}(\hat\rho V) + \frac{\partial}{\partial\zeta}(\hat\rho W)\right) \qquad \text{-----(8)}$$

where,

$$\hat\rho = \rho h$$

$$A_1 = \left(A - \frac{U^2}{a^2}\right)$$

$$A_2 = \left(B - \frac{V^2}{a^2}\right)$$

$$A_3 = \left(C - \frac{W^2}{a^2}\right)$$

A, B and C are terms in the expansion of Laplace's equation. The LHS can be factored using an approximate factorization scheme and becomes

$$L_\xi L_\eta L_\zeta \Delta\phi^* = \text{RHS}$$

RHS involve terms containing fluxes at '*' level and density at $n^{th}$ level all known. We solve for $\Delta\phi^*$. At convergence $\Delta\phi^* \Rightarrow 0$ and $\phi^* \Rightarrow \phi^{n+1}$. In our scheme the spanwise marching is done explicitly. That is, $L_\zeta = 1$. Also, to get an initial guess for $\phi^*$ expansion about $n^{th}$ level is used instead of * level.

Finally, in the wake circulation convection equation is solved using

$$\Gamma_\tau + \overline{U}\,\Gamma_\xi = 0$$

## 3.4 Dynamic Blade Motion

In this section the mechanism to include blade cyclic pitch variation, flapping and elastic deformations is described.

The fundamental task of the comprehensive analysis is the computation of the trim solution. The trim procedure produces control inputs ( cyclic and collective ) for known thrust through an iterative aerodynamic/ dynamic/ elastic computation. This input can either be obtained from another comprehensive load ( CAMRAD ) or from flight test data. A pioneering effort in the coupling of CFD methods to a comprehensive code is shown in Reference 12. In this technique a small disturbance for a full potential code was coupled to CAMRAD as part of the trim procedure. The coupling was achieved by providing a partial angle of attack from CAMRAD to the CFD code and, in turn, returning the blade load to CAMRAD. The partial angles represent geometric twist, blade motion ( flapping and deformation ) and wake inflow effects and were imposed through a transpiration condition at the blade surface.

For efficiency, the CFD computation was performed outside the trim loop. The CFD solution for the lift was specified as a base solution used inside the trim loop. Table lookups were used to provide a correction to the lift and angle of attack. Convergence was achieved when the angle of attack obtained yielded no correction to the base solution.

In the present method, the aerodynamics is treated more accurately, including the time dependent wake, and treats the entire flow field in a unified way without requiring separate wake models and computational boxes around each blade. The CFD calculation, which models the basic three dimensional and unsteady inviscid flow, including the wake, requires a knowledge of the blade deformation and motion to properly predict the loads.

In HELIX-II the cyclic pitch and flapping are included as changes to the blade fixed grids. These changes are confined to a region close to the blade using suitable blending functions. The procedure is described as follows. Blade pitch variation is given by

$$\theta_B = \theta_o + \theta_{1C} \operatorname{Cos} \psi + \theta_{1S} \operatorname{Sin} \psi + \text{higher harmonics} \qquad \text{----} (9)$$

Here, $\theta_0$, is the collective pitch required for a given trim condition

$\theta_{1C}$ , is the Cosine component of the cyclic pitch variation.

$\theta_{1S}$ , is the Sine component, $\psi$ is the azimuth varying in time.

$\theta_0$, $\theta_{1C}$, and $\theta_{1S}$ are obtained either from flight tests or from a comprehensive code.

Each blade sectional grid is rotated by $\theta_B$ using,

$$\overline{X}_{new} = \overline{X}_{old} \operatorname{Cos} \theta_B + \overline{Y}_{old} \operatorname{Sin} \theta_B$$

$$\overline{Y}_{new} = - \overline{X}_{old} \operatorname{Sin} \theta_B + \overline{Y}_{old} \operatorname{Cos} \theta_B$$

Then the 'old' and 'new' grids are blended using:

$$X_{new} = \overline{X}_{new} f_{ij} + X_{old} (1 - f_{ij})$$

$$Y_{new} = \overline{Y}_{new} f_{ij} + Y_{old} (1 - f_{ij})$$

where $f_{ij}$ is blending a function that is 1 near the blade and goes smoothly to zero near the boundaries. This blending process keeps the changes local and leaves the grid near the boundary unchanged. The blade flapping motion is given by

$$\beta(\psi) = \beta_o + \beta_{1C} \operatorname{Cos} \psi + \beta_{1S} \operatorname{Sin} \psi + \text{higher harmonics}$$

where $\beta_0$ is the coning angle , $\beta_{1C}$ and $\beta_{1S}$ are cosines and sine components of the flapping motion corresponding to the first harmonic.

From $\beta(\psi)$ , for each time step a flapping deflection is computed using

$$\Delta Y \Big|_{\text{flapping}} = r \,.\, \sin \beta_{\psi}$$

w^ .re r is the section radius at which the flapping deflection is computed and it is maximum near the tip. This flapping motion is incorporated using

$$\overline{Y}_{ijk\,(\text{new})} = Y_{ijk\,(\text{old})} + \Delta Y \Big|_{\text{flapping}}$$

$$Y_{ijk\,(\text{new})} = \overline{Y}_{ijk}\, f_{i\,j} + (1 - f_{i\,j}) Y_{ijk(\text{old})}$$

Finally, a grid velocities are computed using :

$$\overline{V}_{\text{grid}} = \frac{\overline{X}_{t+\Delta t} - \overline{X}_t}{\Delta t}$$

where $\overline{X}_{t+\Delta t}$ is the grid coordinates at time t + Δt and $\overline{X}$ is the grid at time t. These velocities are added to the physical velocity while solving for the potential.

The blending scheme is shown in Fig. 4. The torsional and bending deformation due to aeroelastic effects can be accounted for using the same technique. Here, instead of specifying the angles, they are obtained from a coupled structural analysis code. This code takes aerodynamic loads as input and returns the blade deformation as output.

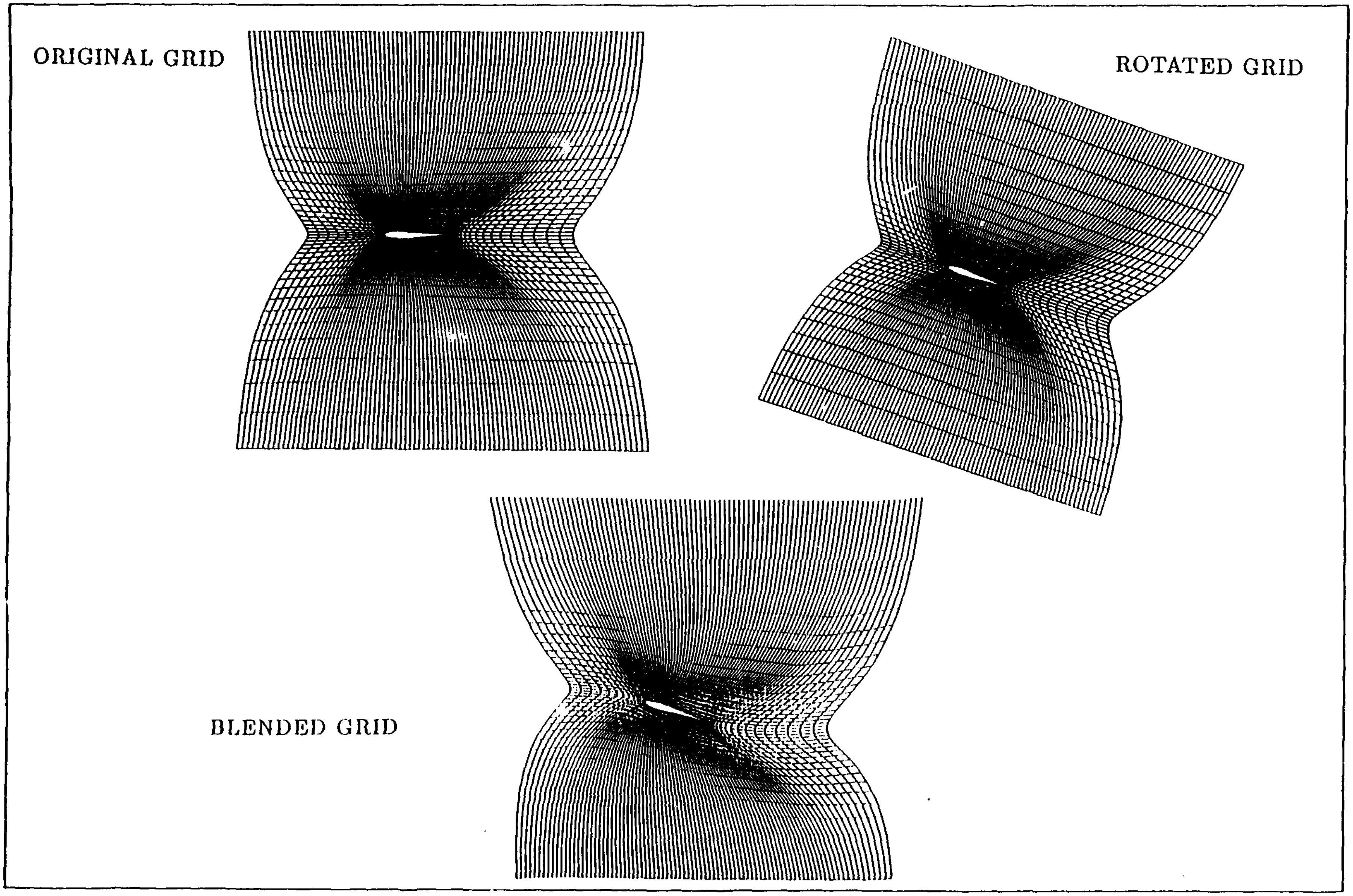


Figure 4: Blending method to incorporate dynamic grid changes in forward flight

# 4. Demonstration of Blade, Vortex Direct Impingement

In a separate effort, HELIX - II code was coupled to a BVI method and is described in Ref. 13. Here, also a vortical velocity computed as a velocity correction and added in the solution to mass conservation equation every time step.

## 4.1 Vortical velocity computation:

Every time step, an accurate, diffusion-free vorticity, denoted $\overline{\omega_0}$, is computed on the Eulerian grid near the blades, using the L field method[13]. This involves convecting the L field coordinates on the Eulerian grid and transferring vorticity coefficient values with a simple low order interpolation scheme. The next operation in each time step involves computing velocities that correspond to this vorticity. For doing this involves a vector potential, $\vec{A}$, in three dimensions ( or stream function in two ) such that

$$\vec{\nabla}^2 \vec{A} = - \vec{\omega}$$

Then the vortical velocity is given by

$$\vec{q}^{\,V} = \vec{\nabla} \times \vec{A}$$

Instead of computing the above three full Poisson solutions, we use a corrected, primitive variable Navier-Stokes approach. This is equivalent to computing one, or a small number of explicit Poisson iterations each time step. In this approach, the velocity components are convected directly on the E Grid ( using, in our case, the image method ). Since the velocity from the previous time step corresponded to the exact vorticity at that same time step, the convected velocities will closely correspond to the new exact vorticity at the new time step. The main difference is that the exact vorticity, denoted $\vec{\omega}_0$, does not exhibit numerical diffusion, since it was computed using the L Field method, whereas the vorticity computed from the convected velocities does. This is due to the diffusion inherent in the direct convection method, whether a standard numerical method or the image technique is used.

The two distributions, $\vec{\omega}$ corresponding to the convected $\vec{q}$, and the exact value, $\vec{\omega}_0$, are expected to differ only in the high frequency components. Accordingly, starting from $\vec{\omega}$, we would expect a simple Point-Jacobi-like ʻorrection step, rather than the full set of Poisson solutions, described above, to be sufficient to correct the velocities so that they correspond to $\vec{\omega}_0$. The correction has the simple form:

$$\delta\vec{q} \;=\; e\vec{\nabla} \times \left(\vec{\omega} - \vec{\omega}_0\right)$$

It can be seen that this is equivalent to a simple Point-Jacobi iteration step in a Poisson Solution. Taking the curl of both sides and using the triple-product form, we have:

$$\delta\vec{\omega} \;=\; -e\vec{\nabla}^2 \left(\vec{\omega} - \vec{\omega}_0\right)$$

Assuming $\vec{\omega} - \vec{\omega}_0$ vanishes outside a small region, and $\vec{\omega}$ and $\vec{\omega}_0$ coincided at the last time step, then a single application of this formula should correct for numerical convection-induced diffusion. In practice, several applications may be required at each time step. The computing time is small, however, since vorticity is non-zero oniy in a small region.

This correction would not be effective for solving for $\vec{q}$ if we did not have a convected velocity to start from. Then, it would be equivalent to completely solving Poisson's equation with a Point-Jacobi method, which would require many iterations since it is very inefficient for long wave length errors. Numerical convection of velocities together with a single or small number of correction steps, however, can be seen to be very effective. In Fig. 5 we display initial vorticity contours of .4 and .8 maximum value computed from a velocity field that is being smoothed to simulate numerical convection-induced diffusion. In Fig. 6a vorticity is plotted along a horizontal line through the center after 0, 50, and 100 diffusion cycles, with no corrections. In Fig. 6b the same values are plotted, but with a single application of the correction method each step, with $\vec{\omega}_0$ set at the initial $\vec{\omega}$. It can be seen that diffusion is effectively eliminated.

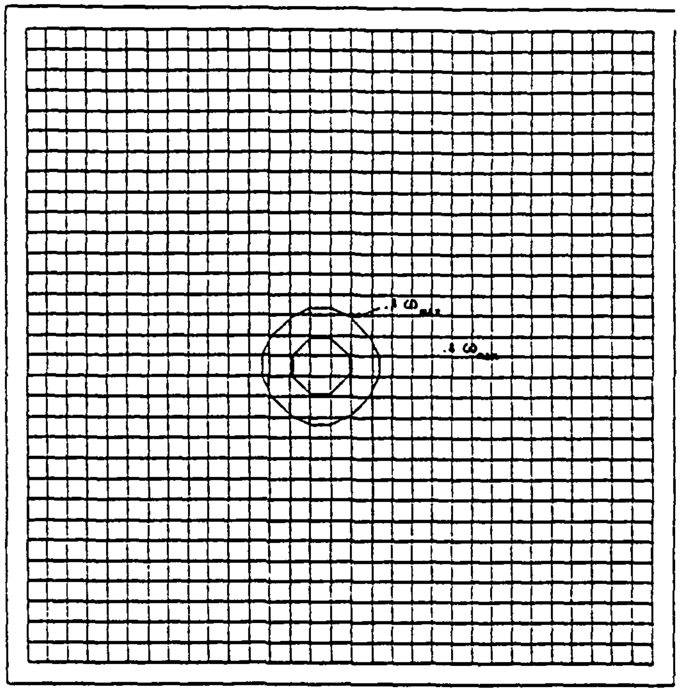


Figure 5: Vorticity matching - initial vorticity contour

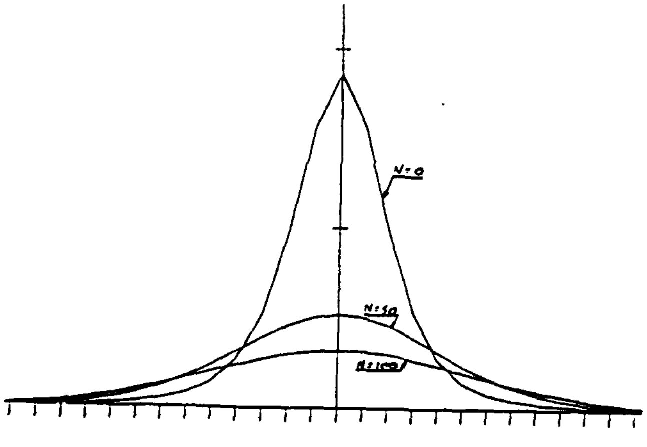


Figure 6a: Vorticity diffusion without vorticity matching

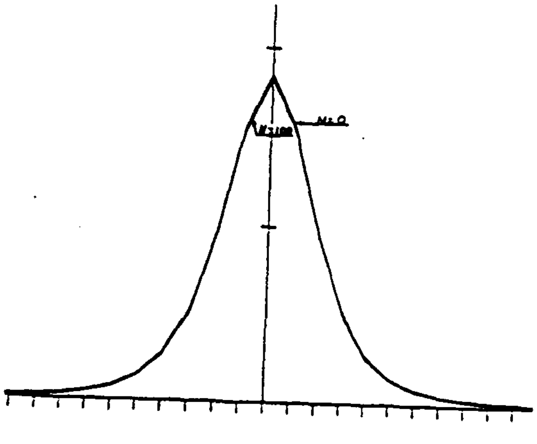


Figure 6b: Vorticity diffusion with vorticity matching

## 4.2 Coarse Grid Region

Away from the blades, where the grid is coarse, this velocity correction takes a particularly simple form : Rather than computing velocity corrections so that $\vec{\omega}$ matches an imposed $\vec{\omega}_0$, we simply drive $\vec{\omega}$ to zero outside a region defined by the L Field coordinates. This process can be described as *vortex capturing.* This capturing technique is simpler to use in coarse grid regions than the $\vec{\omega}$ matching version since it does not involve defining an $\vec{\omega}_0$ distribution on the L Field and computing the stretching factor. Further, even a defined $\vec{\omega}_0$ distribution would have to be scaled if it were to be imposed, or a fine grid required everywhere, since the grid that we should use for computational efficiency can be too coarse in certain regions to resolve the initial distribution. The capturing scheme operates in a similar way to shock capturing schemes, where the discontinuity is automatically spread over several grid cells regardless of grid size with a distribution that is not specified but results from the numeric.

This form of the method is almost completely Eulerian: It only requires a single Lagrangian variable, s, which measures the distance to the vortex centroid. Many types of Lagrangian distance function appear to work: A distance function useful for confining the vortex sheets, in our case, is

$$s = (cjo)^2$$

To accommodate strong concentrated line-type vortices produced by blade-tips, for values of *cko* near *cktip*, this can be modified to

$$s = (cjo)^2 + (cko - cktip)^2$$

where *cktip* is the value of *cko* at the blade tip. Other variables can be used: We have had success with the magnitude of the shed vorticity, which is maximum at the centroid. This is computed shortly after the sheet has been shed and then numerically convected as a passive scalar. Like the other functions, this serves to provide a vector ( the normalized gradient ) in the direction of the centroid of vorticity. Results are presented in Fig. 7 for our HELIX-II code using this latter function, for a general convecting 3 - D wake shed by a single rotor blade. Below, first, the basic formulation of the vortex capturing technique will be given. Then, results of model studies similar to

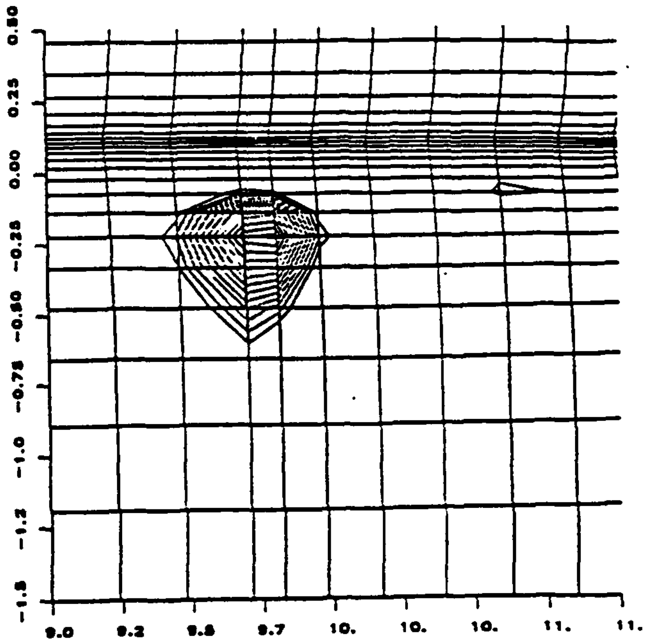


Figure 7: Initial vorticity contour in a cross plane for a single convecting wake sheet obtained from HELIX-I solution

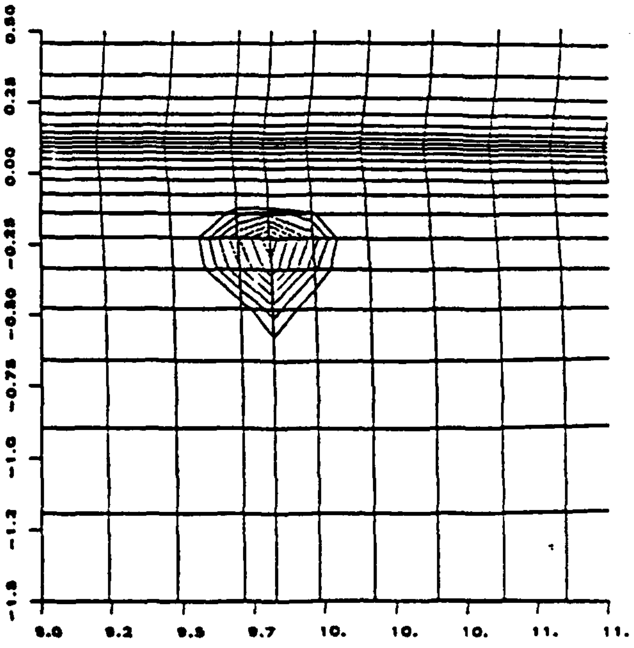


Figure 7b: Vorticity contours after convection without capturing

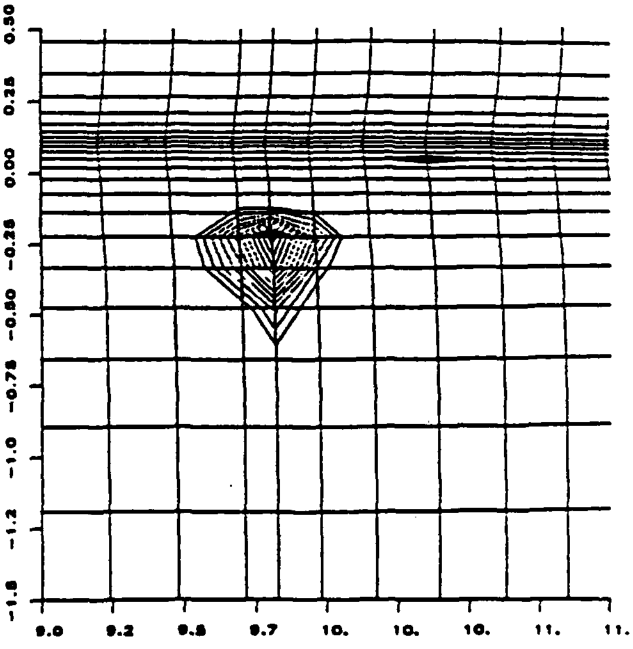


Figure 7c: Vorticity contours after convection with capturing

those presented above for the Near-Blade ( vortex matching ) method will be described. It should be emphasized that this vortex capturing technique is very general - it can also be used near the blades, even for cases with impingement, as long as it is not desired to solve for the detailed, evolving vortex internal structure.

We use a formulation that has a bias towards smaller values of *s*. The correction then transports vorticity towards regions of small *s*, while conserving total vorticity. This has proven to be a robust scheme. It allows concentrated vortices to be accurately and simply convected through regions with both fine and coarse grid cells, without numerical diffusion. If we consider a grid cell with velocity defined at the nodes, then the box-type central differencing that we use to compute the curl results in an $\vec{\omega}$ defined at the cell center ( see Fig. 8 ). We then compute convected values of s at the nodes, and compute ( at the cell center ) a unit vector pointing to the centroid of the sheet :

$$\hat{n} = \frac{\vec{\nabla} s}{|\vec{\nabla} s|}$$

The correction to be added to the velocity is then simply

$$\delta\vec{q} = \varepsilon\, a_l\, \hat{n} \times \vec{\omega}$$

where is a constant relaxation factor and are weights computed at cell nodes ( labeled *l* ) to enforce the biasing. We have had good success with the simple form:

$$a_l' = \min(0,\ s_l - \langle s \rangle)$$

$$a_l = \frac{a_l'}{\sum_l a_l'}$$

This approach was used for computing blade / vortex impingement and detailed results are shown in Ref. 14 - 15.

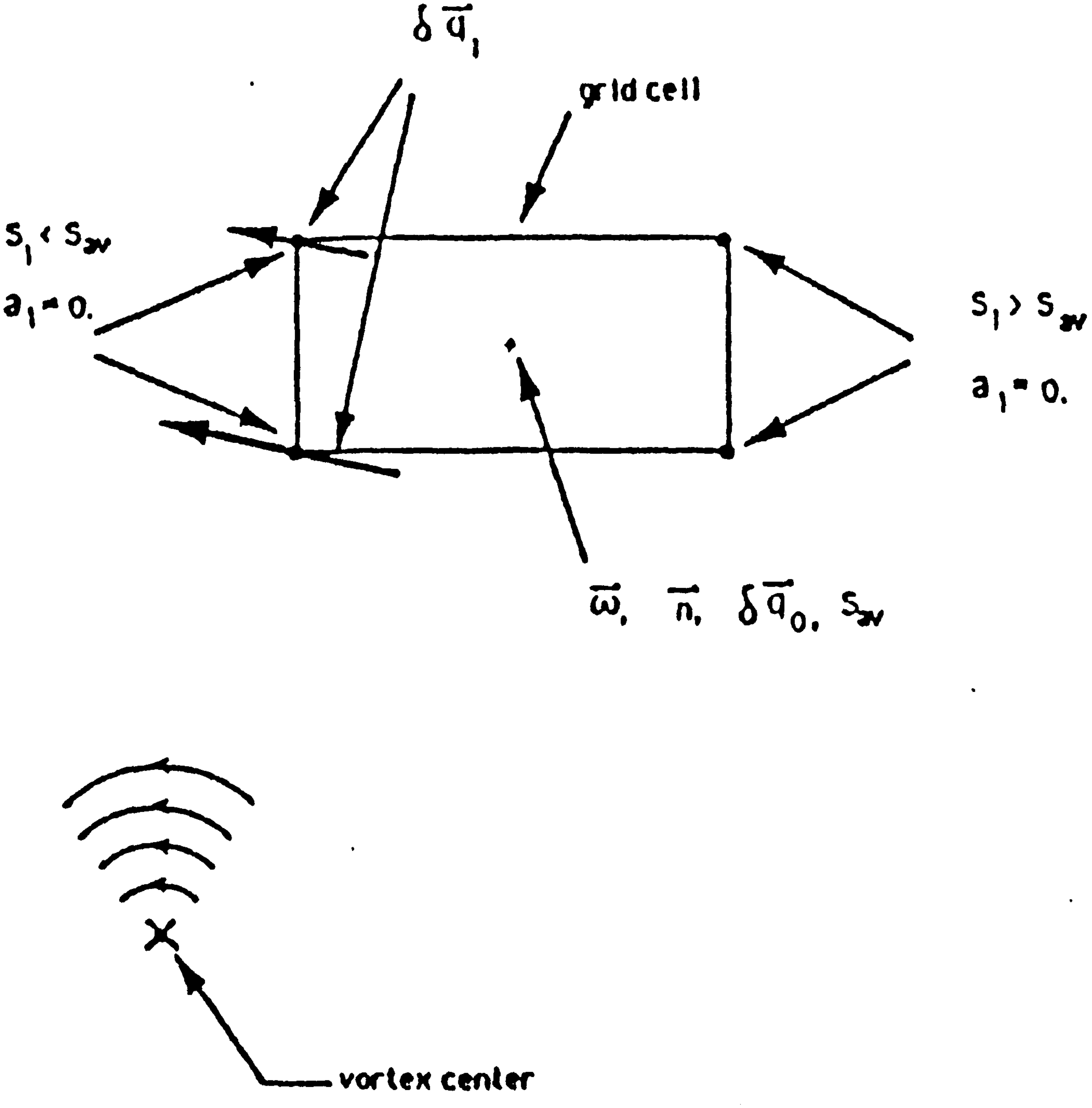


**Figure 8: Schematic of the computation of velocity correction term for vorticity capturing**

## 5. HELIX - II Applications and Results

Before using HELIX -II for forward flight calculations, each module was separated, tested for accuracy. The modules tested are, the unsteady full potential equation solver, the wake convection procedure and finally the code was applied to perform complete forward flight computation at low advance ratios.

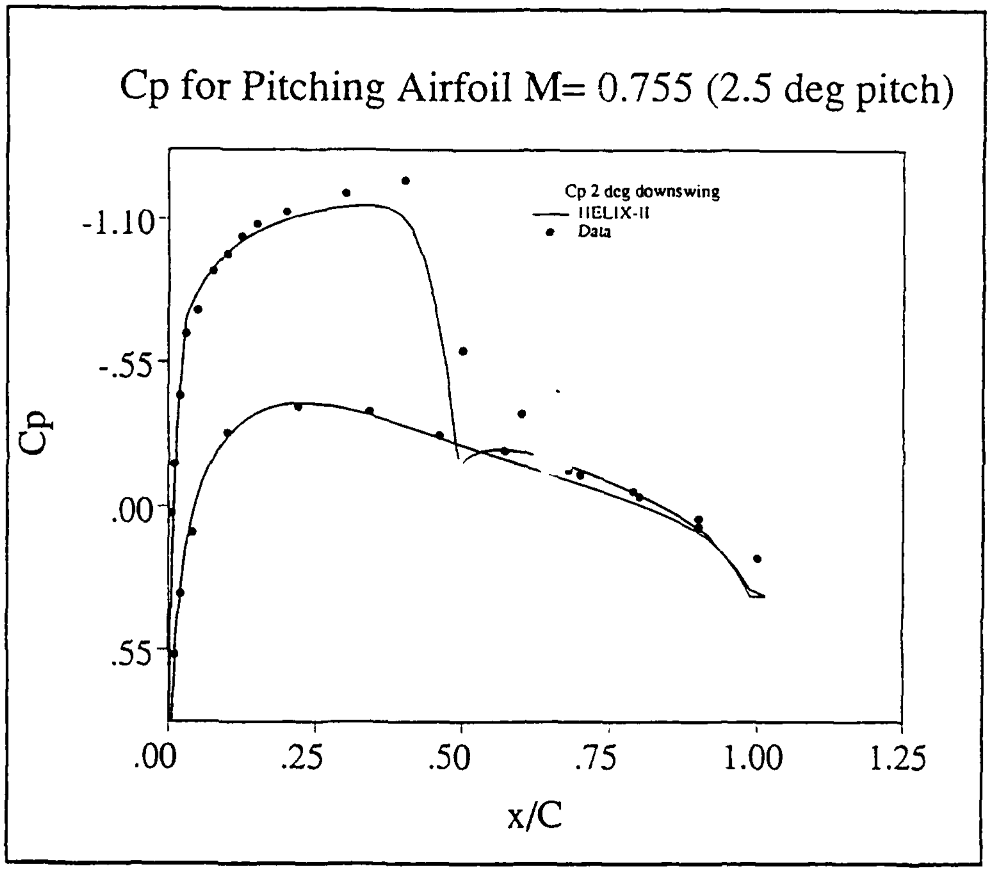
The unsteady full potential solver was applied to compute the time history of loading on an oscillating airfoil for a wide range of Mach numbers. In Fig. 9a the computed lift distribution is plotted against time and compared with experiment . The airfoil oscillates ± 2.5 degrees about a mean of 0 deg. The Mach number is 0.755. Good comparison of the lift history is seen. In Fig. 9b the computed pressure is plotted during a downswing of $2^0$ when the shock is strongest and compared with data and good comparison is seen. Next, the HELIX -II wake convection module was applied for a hover computation on a AH - 1G. The computed load distribution hover solution and wake geometry are shown in Fig. 10.

### Forward flight calculation:

With this background, complete forward flight solutions have been performed on a two-bladed rotor at advance ratios of 0.15 and 0.19. A new H-mesh grid generating code is developed for this purpose. The grid is generated using a two step mapping procedure. First the airfoil coordinates are input at each spanwise station where a radial plane will be defined. In addition to these coordinates, the normalized radial plane location ( normalized by root chord ) and the corresponding twist distribution are input. First a planar H - mesh is generated at each radial station with specified boundaries. The axial boundaries are chosen to be approximately at 1 radius. Next, the planar grids are transformed to a series of cylindrical grids with constant radii. The stream wise extent is determined from the number of blades.

$$\theta_{min} - \theta_{max} = \frac{2\pi}{N_{blades}}$$

The transformation from the planar to cylindrical grid is performed using:



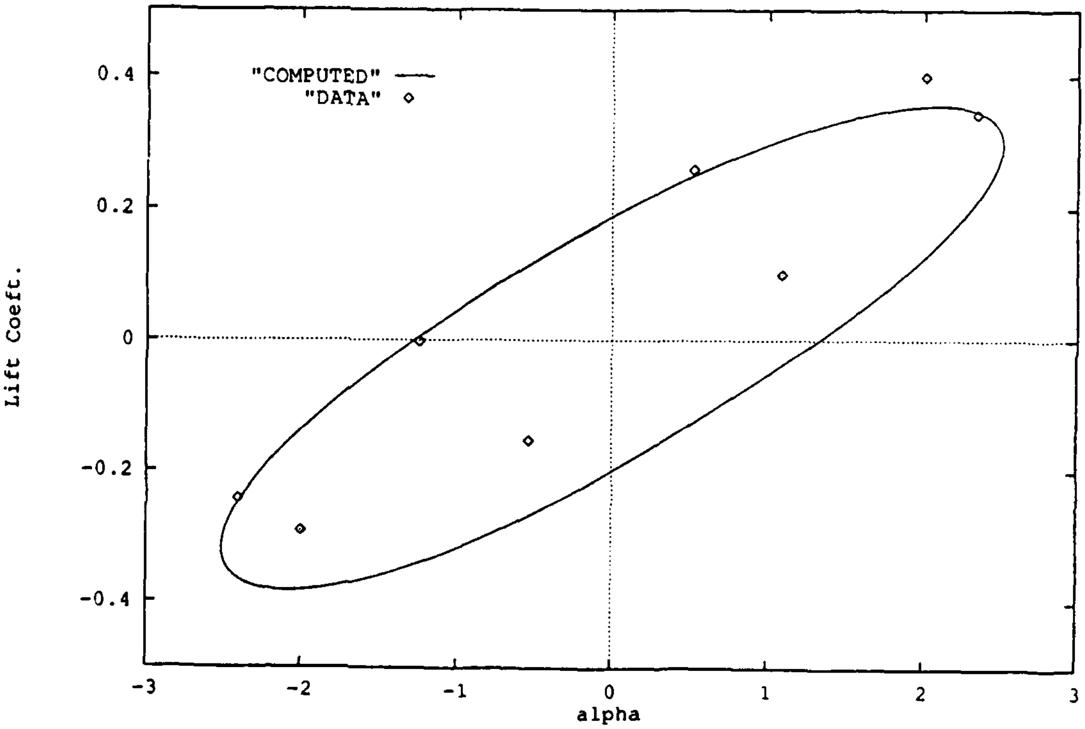


**Figure 9: Comparison of surface pressure and unsteady airloads for an osuillating airfoil at transonic speed - computed using HELIX II**

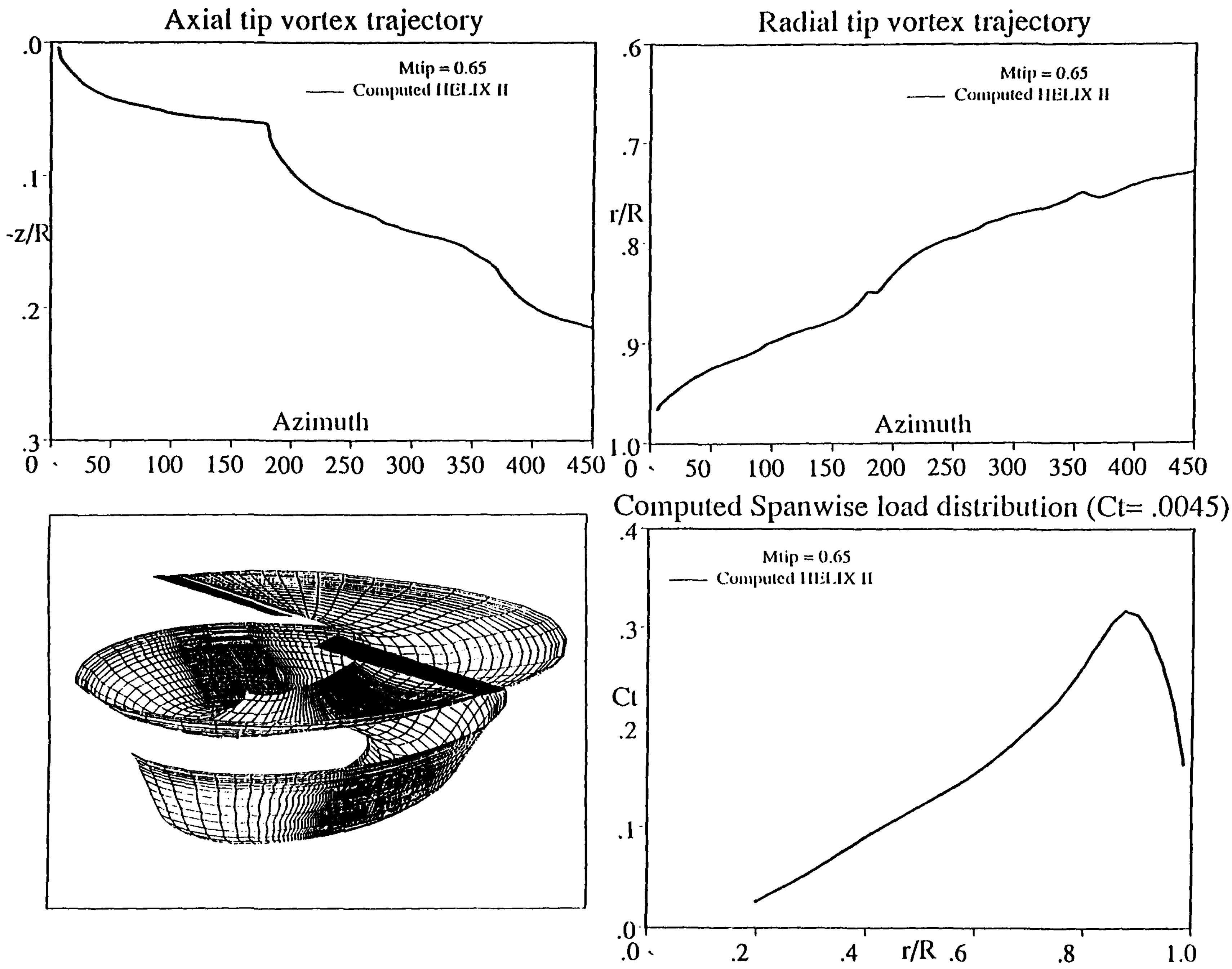


**Figure 10: HELIX-II computed solution for AH-1G rotor, $\mu = 0.0$**

Once the grid is obtained on a single blade it is rotated by appropriate angle to obtain grids on other blades.

The rotor blade chosen for HELIX - II studies has been AH1G rotor with an Aspect ratio of 9.8. It uses a 11% thick symmetric airfoil along the span. Three different grids have been generated containing about 350,000 to 400,000 points. Of this about 200 points are distributed along the azimuth, about 50 in the normal direction and about 40 in the radial direction. A cylindrical section is shown in Fig. 11. The twist and a collective pitch angle are built into the grid by rotating each radial station. Finally, the blade motion harmonics are input from the flight test values. These are the blade coning angle, the first longitudinal and later cyclic pitch coefficients and blade flapping harmonics. These coefficients correspond to a particular trim condition. Two different conditions have been chosen at advance ratios of 0.19 and 0.15. The lateral and longitudinal cyclic pitch variations have been obtained from Ref. 16. Also, the blade flapping harmonics - zeroth ( coning ) and first are input from this report. The tip Mach number is 0.60.

The potential has been initialized to zero. The initial wake is undistorted one obtained with rotational speed and a uniform axial flow. The initial blade position corresponds to 0 degree azimuth. The undistorted wake has 21 nodes distributed along the spanwise direction and 100 azimuthal nodes ( Fig. 12 ). The radial circulation distribution is input from an earlier hover calculation and there is no azimuthal variation of circulation initially. Thus this starting wake is very approximate and hence a better way of starting the solution would involve specifying a more accurate circulation variation from either a lifting a line code or from a previous calculation. In the absence of such information the present starting procedure can be used. This resulrs in a longer computing time - requiring about 6 - 8 revolutions before a periodic solution is obtained. At each time step, the wake is distorted to follow the local flow at that instant and a new node is added to the wake. In the present calculation, for every node added a far wake node is dropped thus maintaining the total number of azimuthal nodes a constant. The effect of the far wake is included using extrapolation.

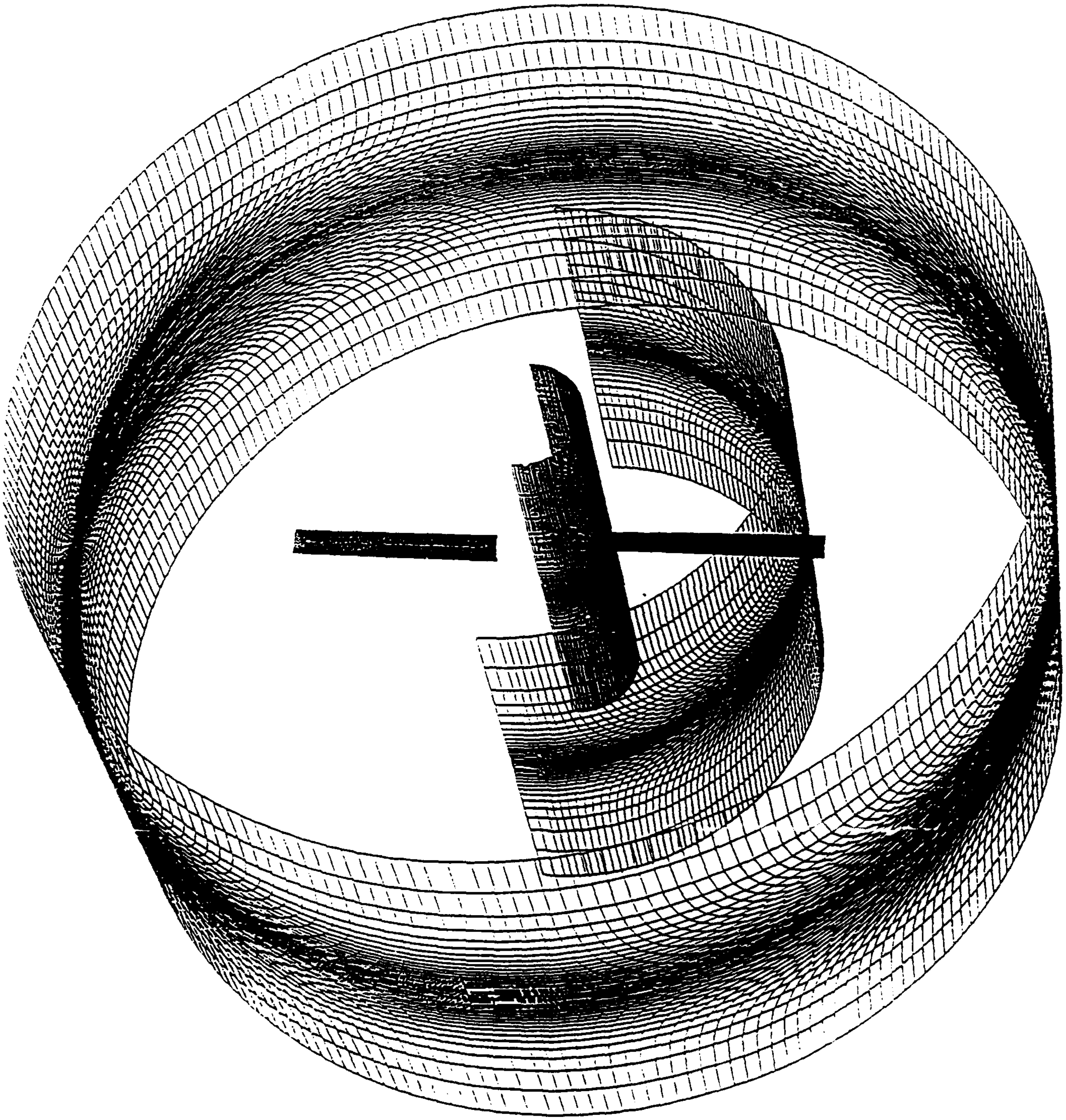


Figure 11: Cylindrical grid used for HELIX -II computations

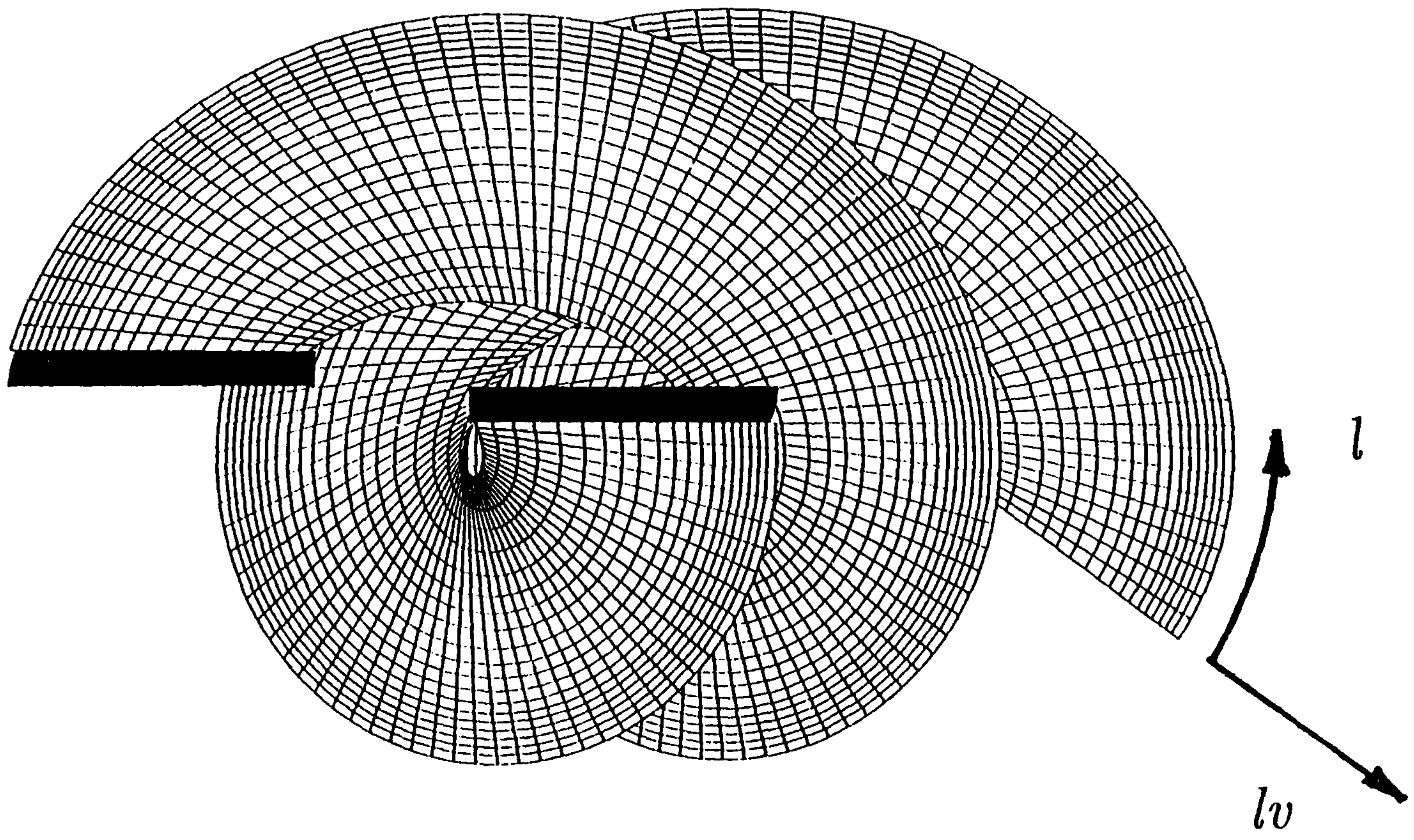


Figure 12: Lagrangian wake coordinate system and the initial undistorted wake used for forward flight computations

In Fig 13 the computed wake geometry is plotted at two different azimuths corresponding to 90 and 180 degrees. This wake is obtained after 6 rotor revolutions of computations. The wake rolls up and convects fairly close to the blade. At 90 degree and 270 degree azimuths their proximity to the blade cause particularly a strong interaction.

Computations have been performed at two advance ratios 0.15 and 0.19. In Fig. 14 contours constant circulation is plotted for 7 rotor revolutions. The loading corresponding to revolutions 5, 6 and 7 are nearly identical signalling the arrival of periodic state. This figure also highlights regions of strong BVI occuring on the retreating side. There is a sudden variation in airloads which begins near the tip and extends over the entire span. In Fig. 15 the computed sectional loads are plotted as a function of azimuth for differnt radial stations. There is a strong BVI induced loading at 90 degree on the advancing side and at 270 degree on the retreating side. In Fig. 16 and 17 the computed load distribution are compared with CAMRAD/JA computations and flight test data(16,17). A detailed analysis of these comparisons loading are made in Ref. 18 and 19. The magnitude of interactions are less severe when compared with the experimental data. This may have been caused by the lack of resolution and hence an excessive smearing of the vortex. In order to prevent this, with this existing wake the solution can be recomputed over a short interval using a finer grid. This interval would cover the region of BVI both on advancing and retreating sides. Also, a smaller time step would enhance the time accuracy. With a finer grid, the core of the vortex and the spreading of the wake will be reduced. With a better definition of the wake region interactions become stronger. The use of a very fine grid for the entire grid will be computationally expensive also may cause wake stability problems. But in the present approach, fine grid is used only as a post processing of the solution over a short interval and since no wake updates are performed on this grid the solution is stable. In order to study the effect of the time step core size and marker resolution computation have been performed using 1 degree time step. The wake for this calculation's interpolated from a 4 degree wake. Also a smaller spreading distance has been used. These computed results are compared with data in Fig. 18. There is a marked improvement in the correlation of peak to peak loading variation due to BVI on the advancing side.

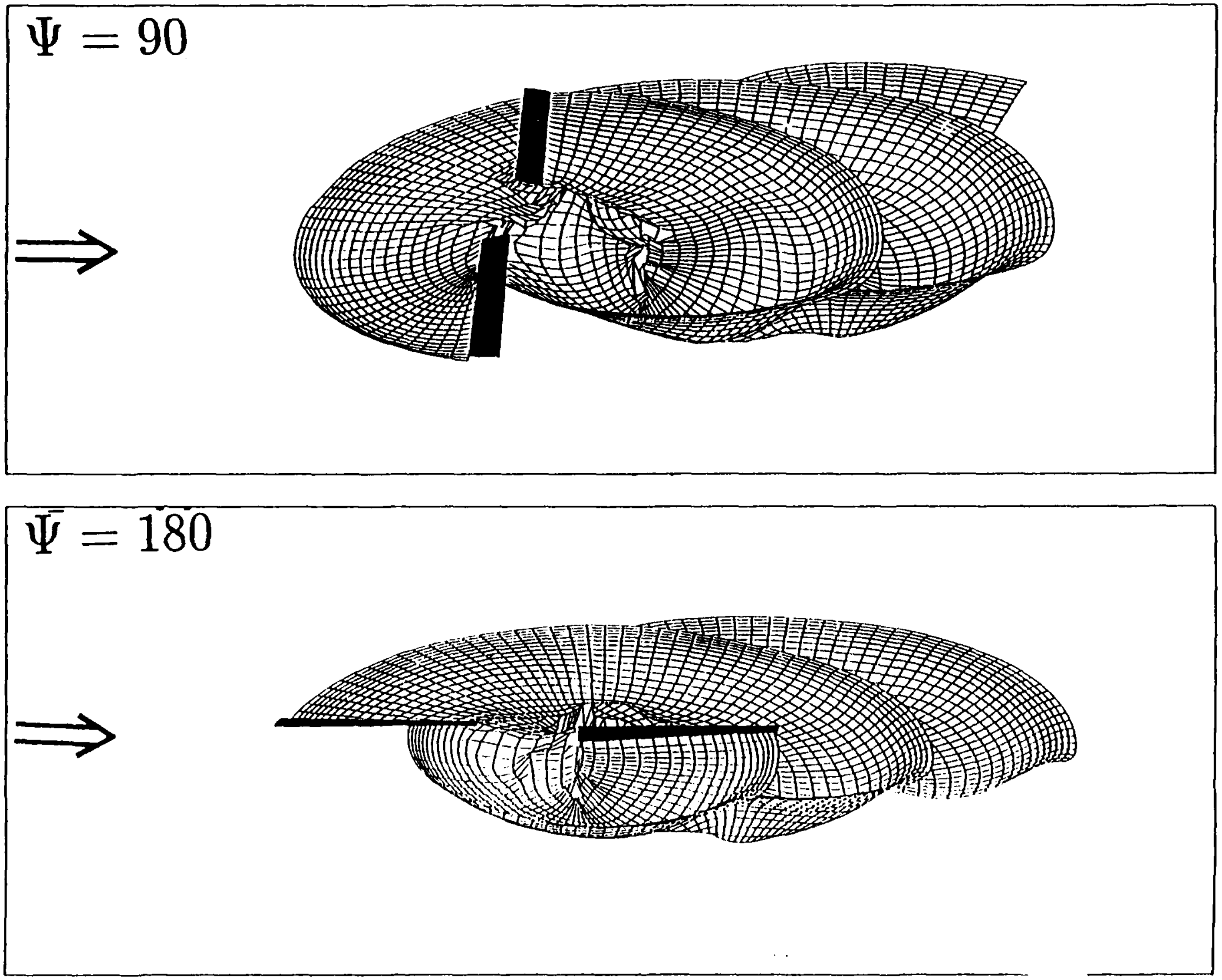


Figure 13: Computed wake geometry at two different azimuths for AH-1G rotor using HELIX-II,$\mu = 0.19$

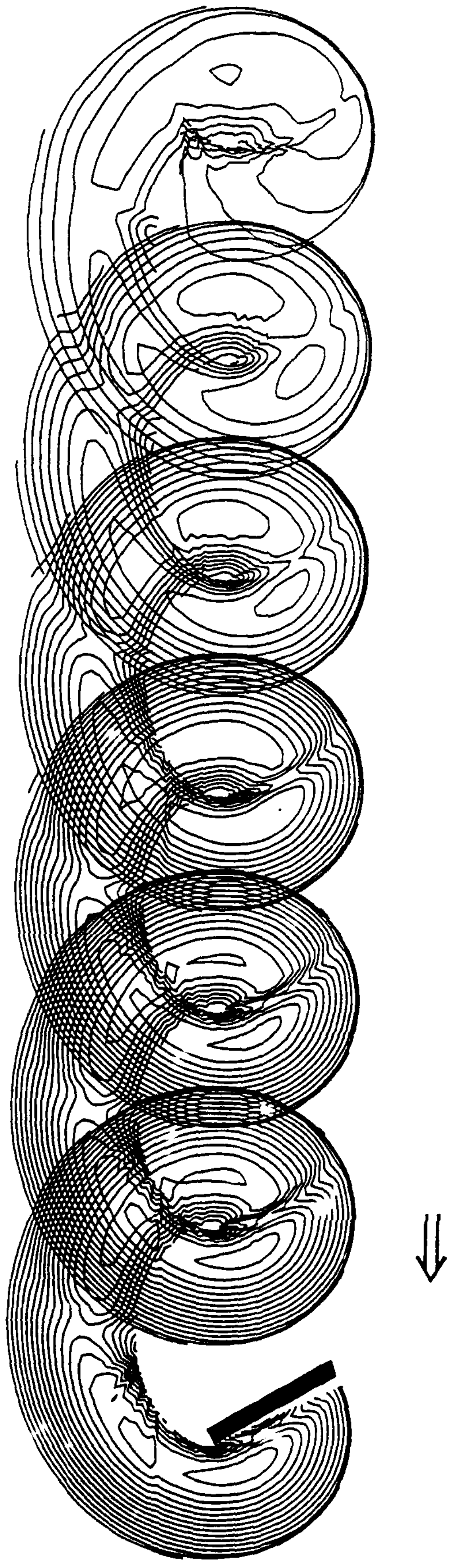


Figure 14: Contours of constant circulation for 7 rotor revolutions, $\mu = 0.19$

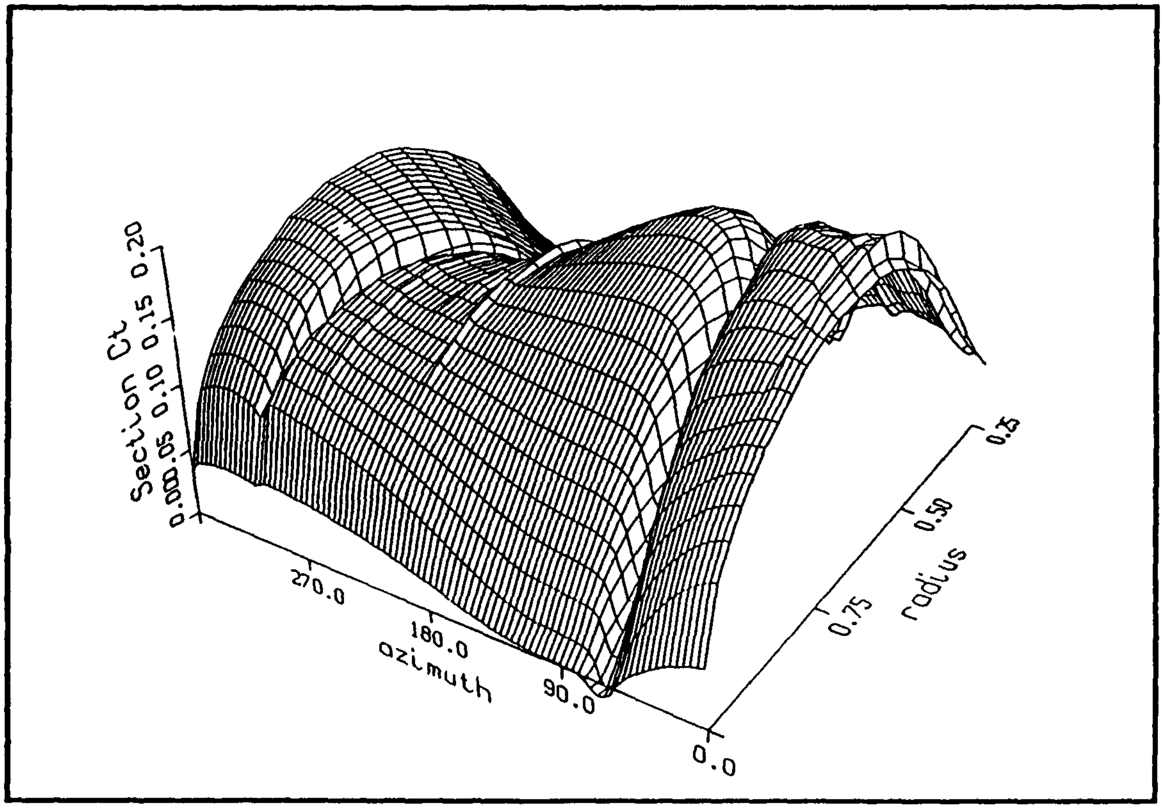


Figure 15: Computed blade airloads for AH1-G as a function of azimuth and rotor radius,$\mu$ = 0.19

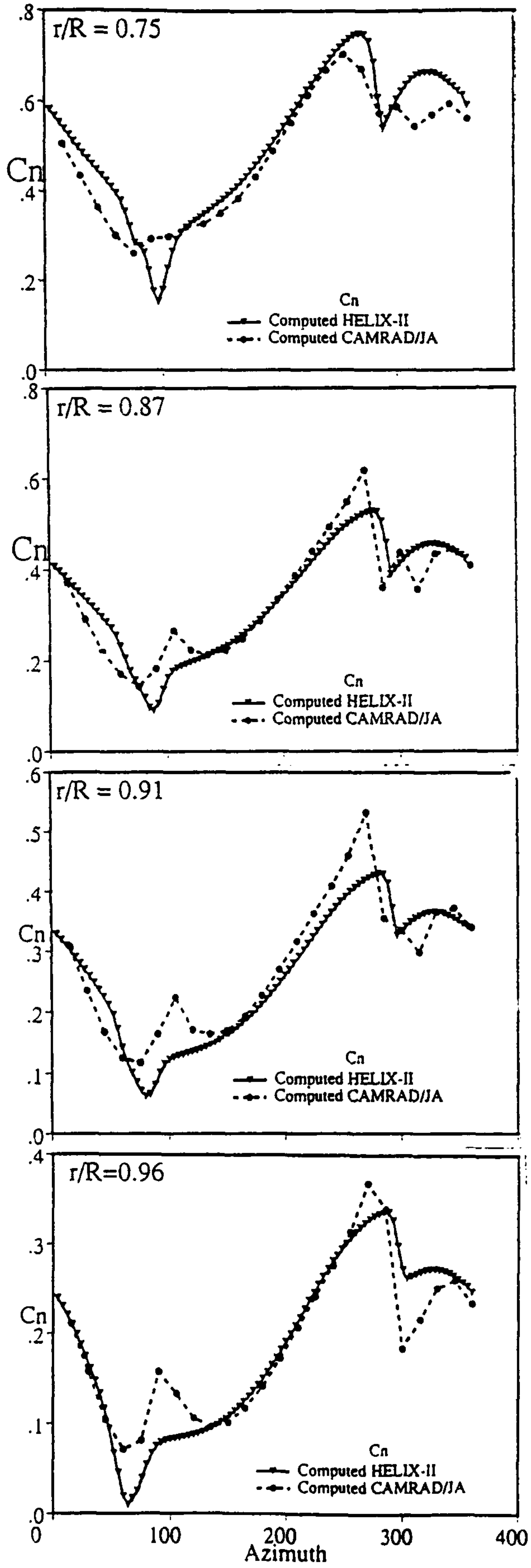


**Figure 16: Comparison of airloads with CAMRAD/JA computations, $\mu = 0\ 15$**

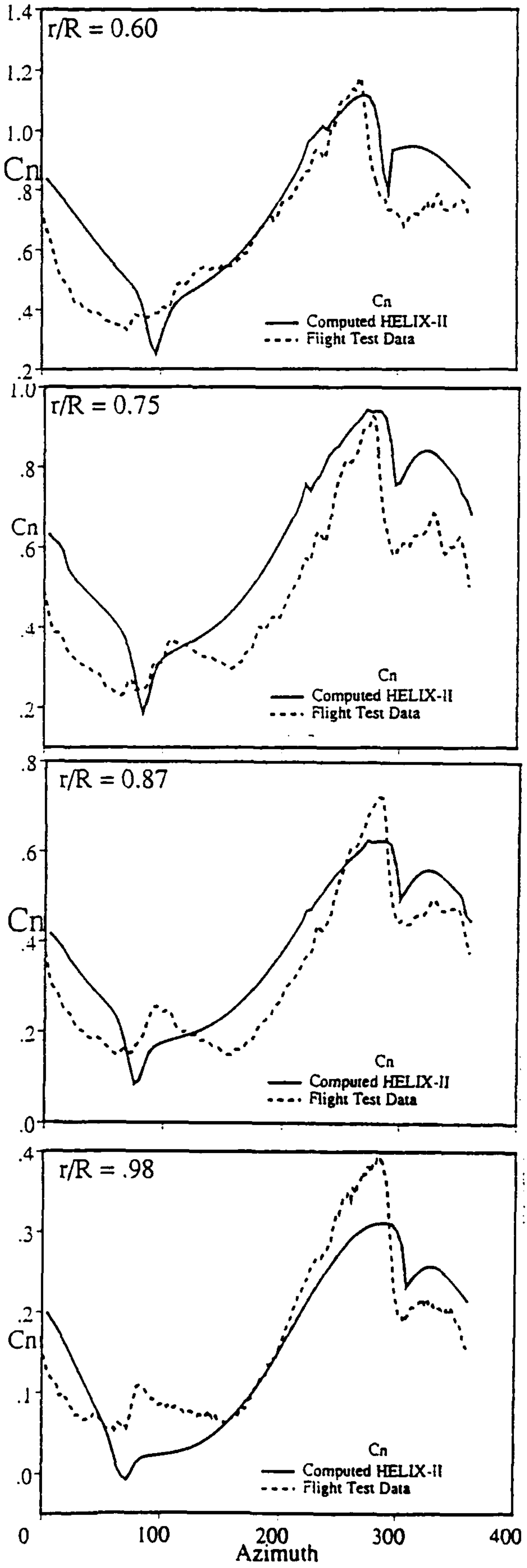


**Figure 17: Comparison of computed airloads with flight test data, $\mu = 0.19$**

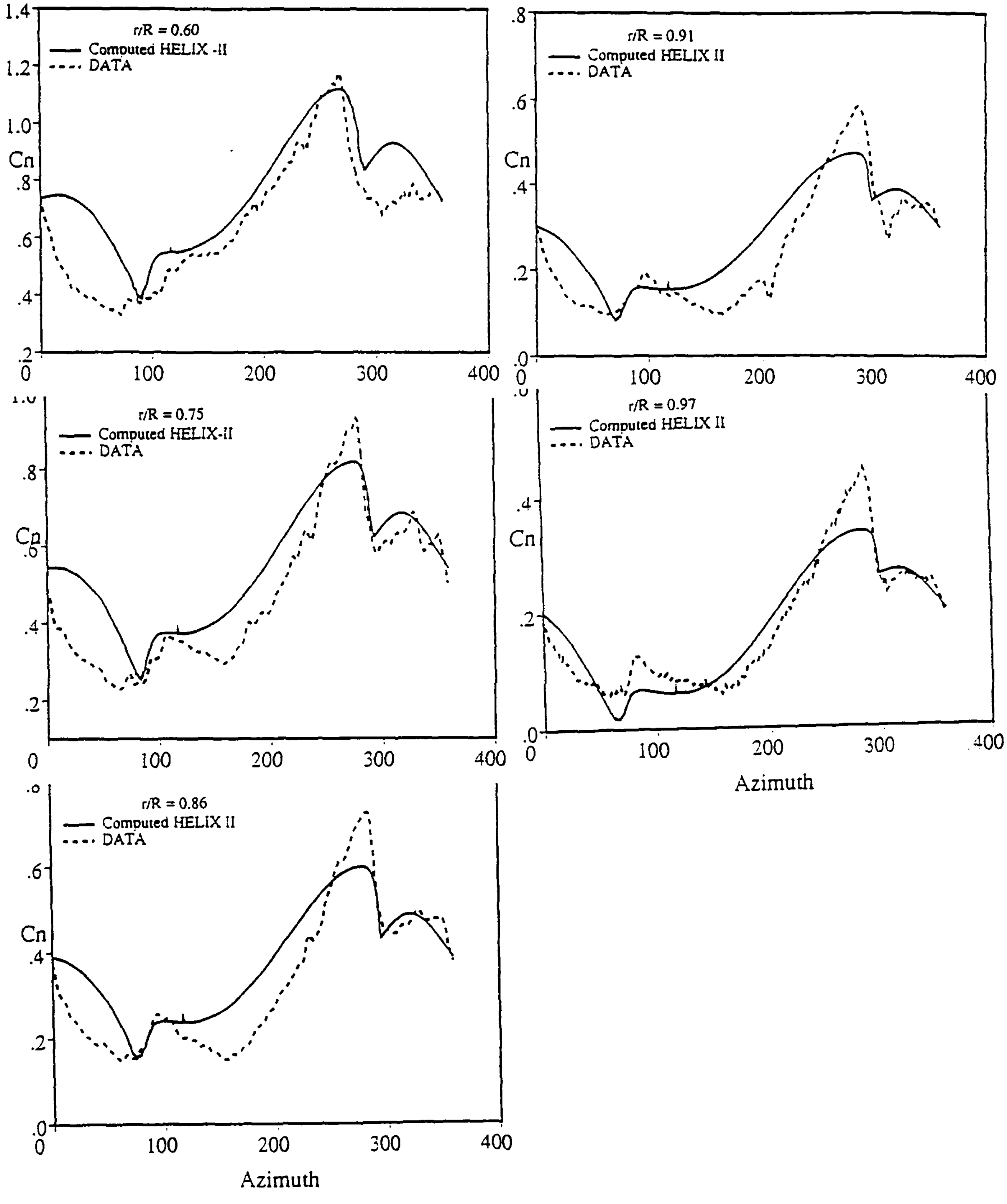


Figure 18: Comparison of airloads for AH-1G using an improved wake model and smaller time step, $\mu = 0.19$

## Conclusions

A new free - wake analysis CFD method has been developed that is applicable both for low and high advance ratio forward flights. At low advance ratios the wake effects are nearly an important as in hover. But at the same time the solution is time dependent unlike hover and hence represent the most difficult region to accurately predict. Several important effects such as a complex wake system, transonic flows near the tip, the specified blade motion and blade deformations due to aerodynamic loads should be taken into account.

HELIX - II handles this complex problem in a unified manner. The wake effects are fully included by the unique vortex embedding procedure. Specified Blade motion terms for a trim calculation are incorporated with simple grid modifications confined to regions near the blade. The method is fully compressible and can capture non - linear transonic shocks accurately. For elastic deformations a coupling procedure with a structural code will be required. When coupled, the aerodynamic loads will be input to the structural module which will in turn provide blade deformations to HELIX - II. The torsional deflections and angles can be incorporated as a grid modification.

At present HELIX - II runs on a super computer at takes about 8 CPU hours on a YMP for 1 trim calculation. With a better starting solution this can be greatly reduced. In addition, the current version performs detailed computation on all blades. Hence requires a large number of grid points especially in the azimuthal direction. This may not be necessary. Detailed computation is required only on one blade and other blades can be represented by a lifting line whose circulation can be obtained from the first blade's circulation at an earlier time. Future efforts will be concentrated in cutting this requirement so that more complex multiple blade configuration can be easily handled. Also, the vortex confinement procedure described earlier can be used for direct interactions.

# BIBLIOGRAPHY

[1] Johnson, W., "A Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics, Johnson Aeronautics Version," Johnson Aeronautics, 1988.

[2] Johnson, W., "Calculation of Blade-Vortex Interaction Airloads on Helicopter Rotors," Journal of Aircraft Vol. 26, No. 5, May 1989.

[3] Rai, M. M., "Navier-Stokes Simulations of Blade Vortex Interact ions Using Higher Order Accurate Upwind Schemes," AIAA Paper 87-0543, Presented at AIAA 25th Aerospace Sciences Meeting, Reno, NV, 1987.

[4] Strawn, R. C., and Barth, T. J., "A Finite Volume Euler Solver for Computing Rotary-Wing Aerodynamics on Unstructured Meshes, " Presented at the 48th Annual Forum of the American Helicopter Society, Wahsington,DC, June 3-5, 1992.

[5] Bridgeman, J. O., Strawn, R., O., Caradonna, F. X., and Chen, C. S., " Advanced Rotor Computations with a Corrected Potential Method, " Presented at the 45th Annual Forum and Technology Display of the American Helicopter Society, Boston, MA May 22-24, 1989.

[6] Ramachandran, K., "Free Wake Analysis of Helicopter Rotor Flows in Hover using a Finite Volume Technique,", Ph.D Thesis, University of Tennessee, Knoxville, 1987.

[7] Ramachandran, K., Tung, C., and Caradonna, F.X., " Rotor Hover Performance Prediction Using a Free-Wake, CFD Method, " Journal of Aircraft, Volume 26, Number 12, December 1989, pp 1105-1110

[8] Steinhoff, J. S., and Ramachandran, K., " The Free Wake computation of Helicopter Rotor Flows in Hover using a Finite Volume Technique, " AIAA Journal , Vol. 28, pp 426, 1990.

[9] Strawn, R. C., and Caradonna, F. X., "Conservative Full Potential Model for Unsteady Transonic Rotor Flows," AIAA Journal, Vol. 25, No. 2, p193, February 1987.

[10] Torok, M. S., and Berezin, C. R., " Aerodynamic and Wake Methodology Evaluation Using Model UH60-A Experimental Data," presented at the AHS 48th Annual Forum, Washington D.C., June 3-5, 1992.

[11] Shankar, V., Ide, H., Gorski, J., and Osher, S., "A Fast, Time-Accurate, Unsteady Full-Potential Scheme," AIAA Journal, Vol. 25, No. 2, Feb. 1987, pp. 230-238.

[12] Strawn, R. C., and Bridgeman, J. O., "An Improved Three Dimensional Aerodynamics Model for Helicopter Airloads Prediction, " AIAA Paper No. 91-0767, Presented at the 29th Aerospace Sciences Meeting, Reno, NV, Jan 7-10, 1991.

[13] Steinhoff, J. S., Yonghu, W., Mersch, T., and Senge, H., "Computational Vorticity Capturing: Application to Helicopter Rotor Flows, " AIAA Paper No. 92-0056, Presented at the 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan 6-9, 1992.

[14] Steinhoff, J. S., Thomas, M., and Yonghu W., "Computational Vorticity Confinement: Two Dimensional Incompressible Flows ," SECTAM XVI Conference, 1992.

[15] Steinhoff, J. S., Senge, H., and Yonghu, W., "An Eulerian/Lagrangian Method for Computing Blade Vortex Interactions, " Journal of Computational Physics, 1991.

[16] Gross, J. L., and Watts, M. E., "Tip Aerodynamic and Acoustics Test," NASA Reference Publication 1179, December 1988.

17 Hernandez, F., and Johnson, W., "Correlation of Airloads on a Two-Bladed Helicopter Rotor, " Presented at the International Specialists Meeting on Rotorcraft Acoustics and Rotorcraft Fluid Dynamics, Valley Forge, PA, OCT 15-17, 1991.

[18] Ramachandran, K., Schlechtriem, S., Caradonna, F. X., and Steinhoff, J. S., " The Application of Vorticity Embedding to the Computation of Advancing Rotor Flows, " to be Presented at the 49th Annual Forum and Technology Display of the American Helicopter Society, St. Louis, MO, May 19-21, 1993.

[19] Ramachandran, K., Schlechtriem, S., Caradonna, F. X., and Steinhoff, J. S., " The Free-Wake Computation of Helicopter Rotor Flowfield in Forward Flight," to be Presented at the 24th AIAA Fluid Dynamics Conference, Orlando, FL, July 6-9, 1993.

# APPENDIX -I

# COMPUTATIONAL VORTICITY CONFINEMENT: A NON-DIFFUSIVE EULERIAN METHOD FOR VORTEX-DOMINATED FLOWS[††]

John Steinhoff,[†*] Clin Wang,[*] David Underhill,[†] Thomas Mersch,[†] and Yonghu Wenren[†*]

## ABSTRACT

A new "Vorticity Confinement" method is described which involves adding a term to the momentum conservation equations of fluid dynamics. This term depends only on local variables and is zero outside vortical regions. The partial differential equations with this extra term admit solutions which consist of Lagrangian-like thin vortical structures (such as vortex "blobs" in 2–D and vortex filaments in 3–D) which convect with a fixed internal structure, without spreading, even if the equations contain diffusive terms. Solutions of the discretized equations on a fixed Eulerian grid show the same behavior, in spite of numerical diffusion.

This modification appears to be very useful in the numerical solution of flow problems involving thin vortical regions. The discretized Euler equations with the extra term can be solved on fairly coarse, Eulerian computational grids with simple low order (first or second) accurate numerical methods, but can still resolve and accurately convect concentrated vortices without spreading due to numerical diffusion. Since only a fixed grid is used with local variables, the Vorticity Confinement method is quite general and can automatically accommodate changes in vortex topology, such as merging.

In this paper, applications are presented for incompressible flow in 2-D, including co-rotating vortices and Vortex Sheet Rollup. The method, however, is not restricted to 2-D (results of an application to 3-D helicopter rotor flow in generalized coordinates have been previously presented).

[††] This work was partially supported by U.S. Army Contracts NAS2–13079, DAAL03–89–C–0027

[†] The University of Tennessee Space Institute, Tullahoma, Tennessee 37388
[*] Flow Analysis, Inc., Tullahoma, TN 37388

## 1. INTRODUCTION.

In high Reynolds number flow, thin regions of concentrated vorticity often exist, which convect through the flow field. These vortical regions can be much smaller in extent than the other length scales in the flow. In these cases the details of the internal structure of these regions may not be of interest, but only the total vortical strength and motion of some suitably defined centroid of each. Computational methods which neglect the accurate computation of the internal structure of these regions can be thought of as a "zeroth order" step in a computational sequence, where the next step would involve, for example, including the effects of viscosity and/or turbulence on this structure.

There are, basically, two ways of treating such "inviscid" flows in conventional computing methods — Eulerian and Lagrangian:

Eulerian methods involve using a fixed computational grid and discretizing and solving the basic partial differential equations which describe mass and momentum conservation in the absence of viscosity (and also include energy conservation for compressible flow). These methods do not require specification of the shapes of the vortical regions: They treat vorticity as being present everywhere and solve the same equations at each point. In computations with these methods, attempts are made to attain a reasonable internal structure for thin vortical regions with a minimum number of grid cells across them. A serious disadvantage concerns numerical diffusion which arises in these computations: After a number of time steps the vortical regions tend to diffuse to much larger sizes than would result from only physical diffusion, unless a relatively large number of grid cells are allocated to the region of concentrated vorticity [1].

A very different approach to solving the same inviscid fluid dynamics problem involves the use of Lagrangian markers that convect with the flow (using some suitably defined mean velocity at each marker location). These methods, in the form of "Vortex Lattice" or "Vortex Blob" techniques for incompressible flow [2] and "Vortex Embedding" methods for compressible flow [3] entail representation of vortex sheets or vortex filaments by surfaces or lines defined by markers. These objects represent the centroids of the vortical regions and the main quantities of interest are the total vorticity around each point of a centroid and its location. Usually, a "spreading" function is specified that, effectively, defines the internal structure of vortical regions treated with this technique. Since this structure is specified, it can be kept constant or varied slowly (to simulate the effects of physical diffusion), thereby avoiding the numerical diffusion problem of Eulerian methods.

Unfortunately, there are disadvantages to these Lagrangian methods that limit their usefulness for many realistic problems: Since the vortical regions are defined by connected sets of markers, the topology of each region should be known beforehand so that a suitable array of markers can be computationally defined. In general flows, multiple sheets can be shed from different places on smooth surfaces and some may reattach, making marker specification very difficult. Further, even in problems with simple vortical regions, if these regions interact with solid surfaces, their topology may change, requiring new specifications of the marker inter-connections. Examples include vortices being "cut" by thin objects, such as wings, and reconnecting. In addition, vortical regions cannot easily be made to merge in a natural way if they are defined by markers. This makes it difficult to compute, for example, merging of vortex rings or trailing vortices generated by aircraft. (Lagrangian methods which use large numbers of unconnected markers with overlapping structures also appear to require some information on the locations of vortical regions for the allocation of markers [4].)

In this paper we present a new method for computing flows with thin concentrated vortical regions. The method uses only a fixed, Eulerian finite difference computational grid and does not involve Lagrangian markers. Hence, like conventional fully Eulerian methods, it does not have the disadvantages of Lagrangian methods. It can treat general concentrated vortical distributions in the form of lines and sheets which are shed from surfaces. These vortical regions can interact with other surfaces and each other and change topology, and no special logic is required. For example, vortex regions can merge, automatically. On the other hand, these thin vortical regions convect with a fixed internal structure, defined over as few as 2-4 grid cells, without spreading, even when the basic finite difference method has significant numerical diffusion.

The method involves adding a term to the momentum part of the basic continuum Euler equations. Even when a diffusion term is also added to these equations, or the basic finite difference solution method has diffusive errors, these modified equations admit solutions which consist of concentrated vortical regions which attain a fixed internal structure and convect without spreading. The extra "Vorticity Confinement" term that is added is local, and simple to discretize. Also, it is only non-zero within the vortical regions, and does not change the total vorticity or mass within those regions. Further, for a large class of vorticity distributions, including those most likely occurring in problems of interest, it does not change the total momentum.

First, the basic method will be described. Then, simple closed-form solutions will be presented for the modified continuum equations. A numerical method for implementing the method in a discretized system will then be given. Finally, examples of the method will be presented for the convection and interaction of concentrated vortical "blobs", and Vortex Sheet Rollup in 2–D. In the conclusion, limitations and possible extensions of the method will be discussed.

## 2. VORTICITY CONFINEMENT METHOD.

2.1. Basic Formulation. Some of the details of the basic method are presented in Refs. [5], [6], and [7]. Diffusion is an integral part of the basic method, and we include it in the continuum equations. (It represents the diffusive part of the numerical error when the equations are discretized.) Thus, we really have a set of modified Navier–Stokes equations. Although the method should be applicable to general compressible flows, we only consider incompressible flow here. We have, in 3–D,

$$\nabla \cdot \boldsymbol{q} = 0 \tag{1}$$
$$\partial_t \boldsymbol{q} = -(\boldsymbol{q} \cdot \nabla)\boldsymbol{q} + \nabla(p/\rho) + \mu \nabla^2 \boldsymbol{q} + \varepsilon \boldsymbol{k}$$

where $\boldsymbol{q}$ is the velocity, $p$ pressure, $\rho$ density, and $\mu$ the diffusion coefficient. For the additional term, $\varepsilon$ is a numerical coefficient which controls the size of the convecting vortical regions. The "Confinement Term" has the form:

$$\boldsymbol{k} = -\widehat{\boldsymbol{n}} \times \boldsymbol{\omega},$$
$$\widehat{\boldsymbol{n}} = \frac{\nabla \eta}{|\nabla \eta|} \tag{2}$$

where

$$\boldsymbol{\omega} = \nabla \times \boldsymbol{q}$$

is the vorticity and $\eta$ is a scalar field that has a local minimum on the centroid of the vortical region. Different versions of the method depend on the definition of $\eta$. In the simplest, described here, we have

$$\eta = -|\boldsymbol{\omega}| \tag{3}$$

(Discretized numerical methods that we have developed to implement this correction are described in Refs. [5], [6], and [7].)

In the confinement term, $\widehat{\boldsymbol{n}}$ is a unit vector pointing away from the centroid of the vortical region and the term serves to convect $\boldsymbol{\omega}$ back towards the centroid as it diffuses away. This convection increases the diffusion term and a steady-state form results when the two become balanced.

Additional possibilities for the method involve specifying an auxiliary field ($\eta^*$) independently of $\boldsymbol{\omega}$, convecting it with the flow, and computing $\eta$ as a function of $\eta^*$. Some discussion of these other versions are provided in Refs. [5], [6], and [7].

The new method has some of the features of the characteristic-based "artificial compression" method of Harten [8]. However, it is much simpler and, unlike that method, the correction is limited to the vortical region.

2.2. Salient Features of Vorticity Confinement Method. It will be seen that steady-state solutions exist (in the frame of the convecting vortex), even with diffusion present, for *any* (positive) value of $\varepsilon$. Our basic point is that it may make more sense to discretized this set of equations (1-3) which have thin, well-behaved vorticity distributions, even in the presence of numerical diffusion, than to discretized the unmodified, inviscid Euler equations which only admit vortical regions that continue to spread, if there is any numerical diffusion.

An important feature of the vorticity confinement method is that the correction is limited to the vortical regions. Unlike artificial viscosity-like terms which are small everywhere except near discontinuities, this correction vanishes outside vortical regions. Another important feature concerns the total change induced by the correction in mass ($\delta I_\rho$) and vorticity ($\delta \boldsymbol{I}_\omega$), integrated over the vortical regions: In general 3–D flow, because of the vanishing of $\boldsymbol{k}$ outside the vortical regions, we have:

$$\delta I_\rho = \varepsilon \int \nabla \cdot \boldsymbol{k} dv = 0$$

$$\delta \boldsymbol{I}_\omega = \varepsilon \int \nabla \times \boldsymbol{k} dv = 0$$

where the integration is done over the vortical regions. Another important quantity that can be conserved with the method is momentum. Here we have a more limited proof. If we have a thin vortical "line" that is slowly varying along its length, then we can take a 2–D section and write for the change in momentum there:

$$\delta \boldsymbol{I}_k = \varepsilon \int \boldsymbol{k} da$$

where the integral is over the 2–D section. In this case:

$$\boldsymbol{k} = \omega \frac{\nabla\omega}{|\nabla\omega|} \times \hat{l}$$

where $\omega$ is the value of $\boldsymbol{\omega}$ in the direction of the vortex line ($\hat{l}$) and $\nabla\omega$ and $\boldsymbol{k}$ are in the 2–D plane of the section. We have

$$\delta \boldsymbol{I}_k = \varepsilon \rho \boldsymbol{J} \times \hat{l}$$

where

$$\boldsymbol{J} = \int \omega \frac{\nabla\omega}{|\nabla\omega|} da$$

In general, $\boldsymbol{J}$ will not be zero. However, for the class of $\omega$ distributions that have two axes of symmetry (such as elliptical distributions) $\boldsymbol{J}$ will vanish due to symmetry. The confinement term is intended to be used where thin vortical regions are convected over relatively long distances and where the velocity (except for that due to the vortex) is slowly varying on the scale of the vortex diameter. In that case, we would expect the viscous terms (either due to the basic numerical convection process or added explicitly) to symmetrize the $\omega$ distribution since any strong, concentrated vortex will be spinning rapidly. As a result, we would expect $\boldsymbol{J}$ and hence $\delta \boldsymbol{I}_k$ to be small. Further, in the context of the above use of the method, where the "external" velocity field is smoothly varying, we should be able to make local corrections to the basic form for $\boldsymbol{k}$ to reduce any non-zero values of $\delta \boldsymbol{I}_k$ that occur due to lack of symmetry. These small corrections could depend on elements of the stress tensor. Other modifications and extensions of the method will be discussed in Section 5.

We will see in the next sub-section that the basic solutions to our modified flow equations are axisymmetric blobs with vorticity that decreases exponentially with radius from the center (in 2–D). Since vorticity is conserved, 3–D vortex "filaments" will have the same structure in each 2-D cross-section. A very important feature of the confinement method, of course, concerns the interaction of these vortices with solid surfaces and with each other. Additional important features concern the roll-up of thin vortex sheets.

Considering first the interaction with solid surfaces, the simplest case involves a viscous flow calculation where the boundary layer is to be resolved in the immediate vicinity of the surface. Then, a fine, high-resolution computational grid will be used in that region. Going back to the basic idea of the method — that it be used only in regions where numerical diffusion would be a problem (i.e., coarse grid regions), it can be seen that the correction should not be used in this high-resolution area and that it be made zero. This can be accomplished by making the coupling constant, $\varepsilon$, depend on grid size so that it vanishes in high resolution areas where it is not needed. This dependence on grid size would also be required to ensure that the confinement correction does not lead to errors in the viscous boundary-layer calculation itself. Other cases, involving the interaction of vortices with surfaces where the grid is not fine and where only inviscid computations are done, have been carried out and show the expected diffusion-free convection. Numerical studies of convection of a concentrated vortex past a cylinder in 2–D are shown in Section 4.1.

The vortex interaction feature can be studied by considering the interaction of vortex pairs. For example, in the high Reynolds number limit, co-rotating vortices that are far apart should stay apart for a relatively long time and ones that are close should quickly merge [9]. The vortices should approximate inviscid flow, except when they finally merge, when there should be a viscous-like behavior. This final merging property is analogous to a Kutta condition for "inviscid" flow separation and an entropy condition in compressible Euler solutions [10]. This feature is necessary for a realistic vortex computation method. Numerical results for two co-rotating vortices are presented in Section 4.2.

The roll-up of a thin vortex sheet with elliptical circulation distribution is a standard test case for vortex dynamics methods. Numerical results of this flow are presented in Section 4.3. An important feature here is the lack of sensitivity of the final main vortex position to the confinement parameter, $\varepsilon$, and that the results reproduce well the salient features of some similar experimentally measured flows.

2.3. Closed-Form Solution: Axisymmetric Vortex. For an isolated axisymmetric vortex in 2–D uniform flow, we have;

$$\eta = -|\omega(r,t)|$$

and

$$\hat{\boldsymbol{n}} = \nabla\eta/|\nabla\eta| = \boldsymbol{r}/r$$

where $\boldsymbol{r}$ is defined with respect to the center of the vortex. We define

$$\boldsymbol{q} = \widehat{\boldsymbol{e}}_\theta T(r,t) + \boldsymbol{q}_\infty$$

where $\widehat{\boldsymbol{e}}_\theta$ is a unit vector in the azimuthal direction and $\boldsymbol{q}_\infty$ is a uniform velocity. Substituting this into the confinement scheme, with simulated numerical diffusion we have, in a frame convecting with $\boldsymbol{q}_\infty$,

$$\partial_t \boldsymbol{q} = \mu \nabla^2 \boldsymbol{q} - \varepsilon \widehat{\boldsymbol{n}} \times \boldsymbol{\omega}.$$

If $\varepsilon = 0$, the solution is

$$T = \frac{T_0}{r}\left(1 - e^{-\frac{r^2}{2\mu t}}\right)$$

This, of course, results in a continually spreading vortical region with radius $\sim \sqrt{2\mu t}$ and no non-trivial steady solution.

When $\varepsilon > 0$, we can write an equation for the steady solution with $\partial_t q = 0$: The solution which is finite at $r \to 0$ is:

$$T(r) = \frac{T_0}{r}\left[1 - \left(1 + \frac{r}{a}\right) e^{-\frac{r}{a}}\right]$$

where

$$a = \frac{\mu}{\varepsilon}$$

is the length scale.

This continuum solution should be a good approximation to the actual solution of the discretized equations with numerical diffusion and the capturing correction, for $r > a$. For $r \lesssim a$, discretization effects will be important since the vortex will be spread over several grid cells.

Other closed-form solutions for confined vortex sheets and some simple numerical tests of the confining method for vortex blobs are presented in Refs. [5], [6], and [7].

## 3. NUMERICAL IMPLEMENTATION.

3.1. Basic Flow Solver. We use an efficient scheme that is, basically, a primitive variable fixed-grid Euler equation solution method: This involves numerically convecting the velocity components and computing a pressure term to enforce mass conservation. The capturing method is then used to compute a correction that, effectively, eliminates the numerical diffusion in the convection. In addition to being very efficient, our method can provide a smooth transition to an unperturbed, conventional primitive variable Navier–Stokes scheme where the grid is fine enough and any turbulence models reliable enough to accurately resolve the flow.

*3.1.1.* First, "convected" velocities are computed on the Eulerian grid.

$$\boldsymbol{q}_1^{n+1} = C\boldsymbol{q}^n$$

In the continuum limit (for small $\Delta t$), this can be written

$$q_1^{n+1} \approx q^n - \Delta t(q^n \cdot \nabla)q^n$$

Any accurate numerical convection routine could be used here. In the results presented in the next section, we use a second order accurate method with second order numerical viscosity.

*3.1.2.* Then, a velocity correction, $\delta q_V$, is made on the grid such that, at each node, vorticity is confined:

$$q_2{}^{n+1} = q_1^{n+1} + \delta q_V^{n+1}$$

The computation of this convection will be described below in 3.2.

*3.1.3. Enforcement of Mass Conservation.* A potential is solved for on the Eulerian grid such that the sum of the gradient of the potential and the convected velocity with correction enforces mass conservation and normal flow conditions on solid surfaces. Our use of convected velocities together with a potential is similar to the split-velocity Euler/Navier–Stokes solver of Ref. [11]. We have

$$q^{n+1} = q_2^{n+1} + \nabla\phi^{n+1}$$

The potential, $\phi^{n+1}$, satisfies the Poisson equation

$$\nabla^2\phi^{n+1} = -\nabla \cdot q_2^{n+1}$$

and normal flow conditions on solid surfaces:

$$\partial_n\phi^{n+1} = -q_2^{n+1}$$

The potential has the effect of a pressure term computed to satisfy mass balance.

$$\nabla \cdot q = 0$$

In the continuum limit, the above steps also satisfy the inviscid momentum balance relation (without the confinement term):

$$\partial_t q = -(q \cdot \nabla)q + \nabla(P/\rho)$$

Any Poisson solver can be used for this step. For application to 3-D helicopter flow, a Jameson-type generalized-coordinate, conservative finite volume method was used on a blade-conforming grid and an efficient implicit Approximate Factorization method was used to converge to a potential solution [12]. The results presented in Section 4.1 use a multigrid solver and in 4.2 and 4.3, an FFT-based and ADI solver.

*3.1.4. Corrector Step.* Velocities computed in the above steps are used as "predictor" values and these steps are repeated in a "corrector" mode.

The above steps are repeated for each time step.

3.2. Confinement Term. The confinement term is added explicitly at each time step. We use a formulation that has a bias towards smaller values of $\eta(-|\nabla\omega|)$. The correction then transports vorticity towards regions of small $\eta$, while conserving total vorticity. This has proven to be a robust scheme. If we consider a grid cell with velocity defined at the nodes, then the box-type central differencing that we use to compute the curl results in an $\boldsymbol{\omega}$ defined at the cell center. We then compute average values of $\eta$ at the nodes, and compute (at the cell center) a unit vector pointing away from the centroid of the vortex:

$$\hat{n} = \frac{\nabla\eta}{|\nabla\eta|}$$

The correction to be added to the velocity is then simply

$$\delta \boldsymbol{q}_2 = -\varepsilon \Delta t a_l \hat{n} \times \boldsymbol{\omega}$$

where $\varepsilon$ is a relaxation factor (which can depend on grid cell dimensions) and $a_l$ are weights computed at cell nodes (labeled $l$) to enforce the biasing. We have had good success with the simple form:

$$\begin{aligned} a_l' &= \min\,(0, \eta_l - <\eta>) \\ a_l &= \frac{a_l'}{\sum_l a_l'} \end{aligned}$$

where $<\eta>$ is the averaged value of $\eta$, over the cell nodes.

## 4. NUMERICAL RESULTS.

The basic numerical solution method is second order accurate and incompressible. It is similar to that of Ref. [11]. The method is explained also in Refs. [5], [6], and [7], where an application of the confinement method in 3–D in generalized coordinates is also described for Helicopter Rotor Flow. Applications of the method to convection of a vortex in 2–D uniform flow are also described in Ref. [7]. In this section results will be presented for several 2-D flows.

4.1. Vortex-Surface Interaction. Results of a vortex convecting closely past a cylinder in a flow that is uniform in the far field are presented. In Fig. 1 contour plots of vorticity are presented for a sequence of times. The contours have the same value in each plot, extending from about 30% maximum value, and show that the vortex does not diffuse. For this run, a value of 0.02 was used for $\varepsilon$. Plots corresponding to the first two times are shown in Fig. 2 with $\varepsilon = 0$ (no confinement). The large effects of the numerical diffusion inherent in the basic numerical method are obvious. The grid used in both computations is shown in Fig. 3.

4.2. Co-Rotating Vortices. Two vortex "blobs" are treated, each spread over several grid cells. Neumann conditions are imposed on the boundaries of a 64 × 64 Cartesian grid. Three solutions are presented: one with no confinement and two with confinement with $\varepsilon$ = 0.05. We define a "core diameter". of a vortex by the diameter of its vorticity contour corresponding to about 30% of the maximum value, or about 5 grid cells for the confined vortices. The first two solutions involve an initial separation of 4 core diameters: one without the confinement term and one with. Contours of the initial vorticity distribution are presented in Fig. 4. After 40 time steps of the computation, the contours are plotted in Fig. 5 for the case with no confinement correction. This total time corresponds to a rotation of the two vortices of about .45 radians. It can be seen that numerical diffusion results in extensive spreading of the vortices, since the maximum value of vorticity has decreased by a factor of about 22. The CFL number is about 4. After 80 time steps, or a rotation of about .9 radians, the vorticity contours, as seen in Fig. 6, are spread widely and the maximum vorticity reduced by a total factor of about 33. Results, with the confining correction turned on, at 80 and 320 time steps or .9 and 3.6 radians, are presented in Figs. 7 and 8 respectively. After a few initial time steps, each vortex reaches a stable state with maximum vorticity about .38 of the initial value. This does not reflect a loss of total vorticity but just a redistribution. A third case is presented with the same conditions, but an initial separation of only 2 core diameters. It can be seen that after a rotation of about 1.8 radians the vortices are fully merged, as shown in Fig. 9. This is to be expected for actual vortex blobs [9].

An interesting feature concerns the lower levels of vorticity. Contours starting at 3% maximum in intervals of 3% maximum are presented in Fig. 10 for a case with an initial separation of 3 core diameters, on a 128 × 128 grid. After 1.8 radians rotation, the two vortices show spiral arms similar to those seen in inviscid high-resolution pseudospectral computations of elongated vortices [13]. Although constant-vorticity, co-rotating vortex blobs are expected to have stable solutions if they are far enough apart, the vortices treated here have a smoothly decreasing distribution and should tend to merge with shedding of spiral arms, as in the vortices treated in Ref. [13]. The effects of our confinement term as well as the diffusion are along the gradient of the vorticity magnitude and do not seem to interfere with the shedding of these arms, which are caused by a varying velocity field mainly normal to this gradient. It can be seen that a weak Raleigh-Taylor-like instability causes the arms to break up into small, regular strings of vortex blobs. This could probably be eliminated by a change in the confinement term, so that continuous arms are shed, but it apparently would not have much effect on the overall solution. A simulation starting from almost the same initial vorticity distribution was performed with a vortex-in-cell method with several thousand point vortices on a 128 × 128 grid [14]. The results are plotted in Fig. 11 after a time similar to that for the above run. The outer-most point vortices of Fig. 11 correspond to the outer-most contours of Fig. 10 and the results are quite similar. This agreement was not expected, since the main objective of the confinement method was to only prevent the main vorticity from diffusi‥g and not to accurately treat detailed, low level features of the internal structure.

4.3. Vortex Sheet Roll-Up. This case involves an initially flat vortex sheet in 2-D with an elliptical circulation distribution:

$$\Gamma(S) = \Gamma_0(1 - S^2)^{\frac{1}{2}}$$

in a square domain, where the left wall corresponds to $S = 0$ and the end of the sheet, at $S = 1$, is initially in the center of the domain. In the computation, Neumann conditions are imposed on the four sides of the domain. Initially, the sheet is slightly spread in the vertical direction on the (128 × 128) grid, as shown by the vorticity contours in Fig. 12. The same contours are presented in Fig. 13 for a solution after 280 steps. A value of .0005 was used for $\varepsilon$ (the normalization of $\varepsilon$ is different for the Roll-up case than the cases presented above). A region of concentrated vorticity can be seen to develop. Contours for the same time but with no confinement, shown in Fig. 14, show a considerably more spread vortex.

A final case concerned variations of the solution with the one parameter in the method: $\varepsilon$. In order for our method to be useful the position of the vortex should not depend on $\varepsilon$, only its core size. This is similar to shock capturing methods where it is important that the shock position not depend on the parameter multiplying the artificial viscosity, but only the shock thickness. The same case presented in Fig. 13 after 280 time steps is presented in Fig. 15, but with a value of $\varepsilon$ four times as large (.002). It can be seen that the vortex is much more concentrated but that it is in the same position, to plotable accuracy.

A delta-wing sheds a vortex sheet from the leading edge that is similar in some respects to this rolling-up sheet. Experimentally measured vorticity contours [15] are shown in Fig. 16 (reproduced from that paper) for a sharp-edged delta wing in a cross-stream plane at .3 chord. Also shown in this figure are the Navier-Stokes 3-D finite-difference solutions by the authors of Ref. [15], using a conventional method but, (in one case), with an embedded fine-grid. The resolution near the vortex was finer in their calculations than those presented here (in the cross-stream plane). It is interesting that the computed contours are very similar to our case without confinement, and that the experimental contours are very similar to our case with confinement. Of course the delta-wing is very different from the simple 2-D roll-up treated here, but the salient features of the rolling-up vortex are similar.

## 5. CONCLUSION.

A method has been presented for computing flows with thin, concentrated vortical regions, which should be important for many high Reynolds number aerodynamic flows. Discretized mass and momentum conservation equations are solved on a fixed Eulerian grid, as in conventional Euler/Navier–Stokes methods. However, a "Vorticity Confinement" correction is applied to the momentum conservation equations in the vortical regions.

The effect of the Vorticity Confinement term is to confine concentrated vorticity to thin regions extending over a small number of grid cells as they convect through the flow. The internal structure of these vortical regions attains a fixed, steady-state form without spreading, even though the basic, discretized momentum equations involve numerical diffusion.

Applications of the method to incompressible flows involving inviscid vortex-surface interactions, co-rotating vortices and vortex sheet roll-up in 2-D were presented. These show the effectiveness of the method even when coarse computational grids are used. For co-rotating vortices, at large scales, the vortices act like inviscid solutions. However, at small scales, when the vortices finally merge, salient features of viscosity are automatically simulated. Also, spiral arms, seen in much higher order, more detailed calculations are computed. Finally, the vortex sheet solutions showed the lack of sensitivity of the final vortex position to the confinement parameter, at least for that flow, and they showed that the salient features of experimentally measured rolling-up vortex sheets should be resolved.

The method has already been applied to a realistic helicopter rotor flow in 3-D. However, additional testing is required for more complex flows and for applications to compressible transonic flows. For example, the use of general non-isentropic, compressible flow solvers may require an additional entropy confinement term to avoid entropy diffusion away from concentrated vortical regions. Also, additional testing is required for 3–D applications, including interactions of vortices with solid surfaces. Further, a characterization of the numerical errors should be given for different basic flow solvers with the confinement term.

Interesting extensions include the simple possibilities of having vorticity-dependent upper and lower cut-offs for the coupling constant, $\varepsilon$. These should, respectively, accommodate "waterbag" constant-vorticity models and smoothly varying background vorticity distributions. Further extensions could include extra terms to reduce any numerical errors that are discovered in applications of the method, or to "encode" desired features of the internal dynamics of the simulated vortices.

## REFERENCES

[1] Rai, M. M., "Navier-Stokes Simulations of Blade-Vortex Interaction Using High-Order Accurate Upwind Schemes," *AIAA 87-0543*, 1987.

[2] Hoeijmakers, W. M. and Vaatstra, W., "A Higher Order Panel Method applied to Vortex Sheet Roll-Up," *AIAA Journal*, Vol. 21, 1983.

[3] Steinhoff, J. and Ramachandran, K., "Free Wake Analysis of Compressible Rotor Flows," *AIAA Journal*, Vol. 28, 1990.

[4] Landsberg, A. and Murman, E., "Control of Numerical Diffusion in Computational Modeling of Vortex Flows, 1991.

[5] Steinhoff, J., Contract Reports, 1990, Contracts NAS2–13079, DAAL03–89–C–0027.

[6] Steinhoff, J., Senge, H., and Yonghu, W., "Computational Vortex Capturing," Submitted to J. Comput. Phys., 1991.

[7] Steinhoff, J., Yonghu, W., Mersch, T., and Senge, H., "Computational Vorticity Capturing: Application to Helicopter Rotor Flow," *AIAA 92-0056*, 1992.

[8] Harten, Amiram, "The Artificial Compression Method for Computation of Shocks and Contact Discontinuities: III. Self-Adjusting Hybrid Schemes," *Mathematics of Computation*, Vol. 32, No. 142. April 1978.

[9] Overman, E. A. and Zabusky, N. J., "Evolution and Merger of Isolated Vortex Structures,"*Journal of Physical Fluids*, Vol. 25, 1982.

[10] Harten, A., Private Communication.

[11] Kim, J. and Moin, P., "Application of a Frictional-Step Method to Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 59, 1985.

[12] Steinhoff, J., Ramachandran, K., and Suryanarayana, K., "The Treatment of Convected Vortices in Compressible Potential Flow," *Proceedings*, AGARD Symposium on Aerodynamics of Vortical Type Flows in Three Dimensions, Rotterdam, Netherlands, 1983.

[13] Melander, M., McMilliams, J., and Zabusky, N., "Axisymmetrization and Vorticity-Gradient Intensification of an Isolated Two-Dimensional Vortex Through Filamentation,". *Journal of Fluid Mechanics*, Vol. 178, 1986.

[14] Ma, H. Y., Private Communication, 1992.

[15] Agrawal, S., Barnett, R., and Robinson, B., "Numerical Investigation of Vortex Breakdown on a Delta Wing," *AIAA Journal*, Vol. 30, 1992.

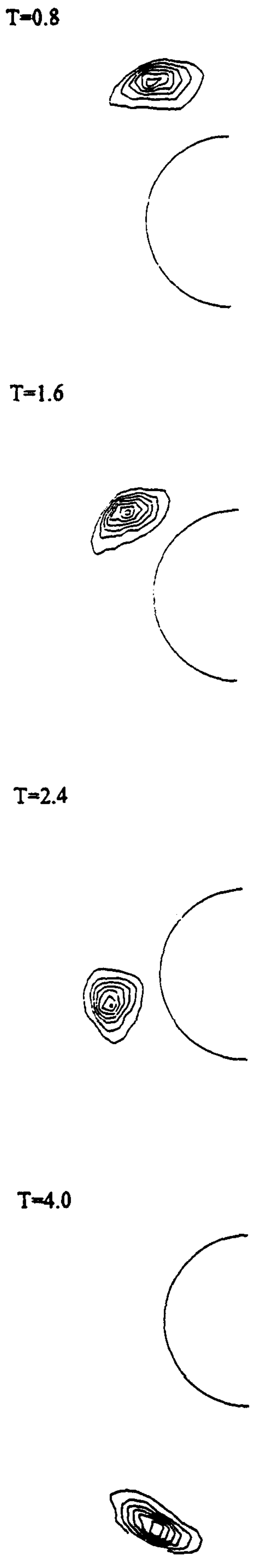


Figure 1. Vorticity Contours, with Confinement

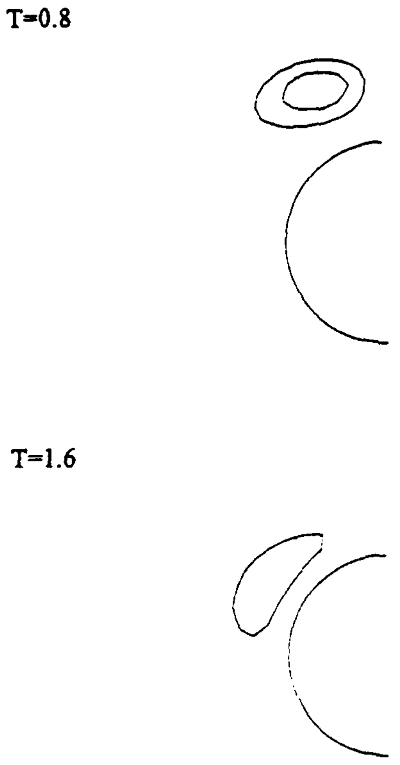


Figure 2. Vorticity Contours, without Confinement

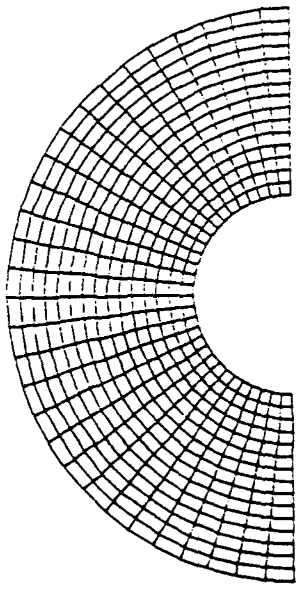


Figure 3. Grid Around Cylinder

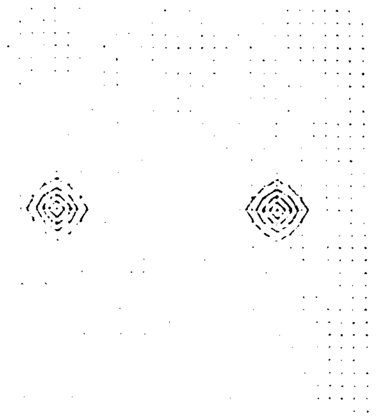


Figure 4. Initial Vorticity Contours

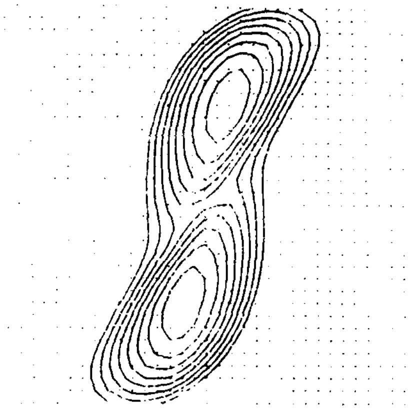


Figure 5. Vorticity Contours after 40 Steps,
4 Core Separation, Without Confinement

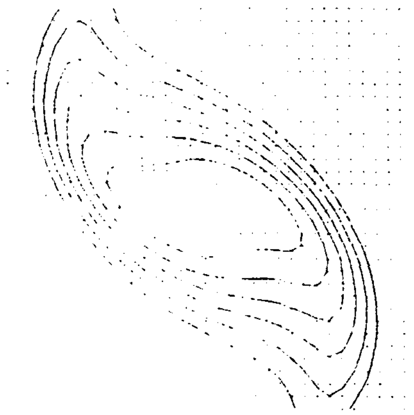


Figure 6. Vorticity Contours after 80 Steps,
4 Core Separation, Without Confinement

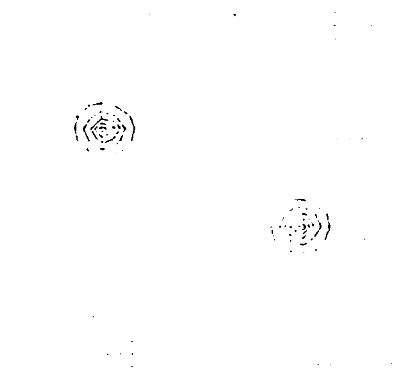


Figure 7. Vorticity Contours after 80 Steps,
4 Core Separation, With Confinement

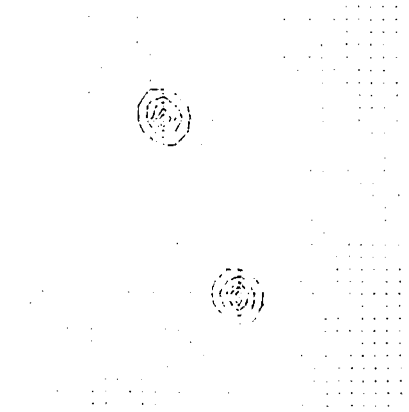


Figure 8. Vorticity Contours after 320 Steps,
4 Core Separation, With Confinement

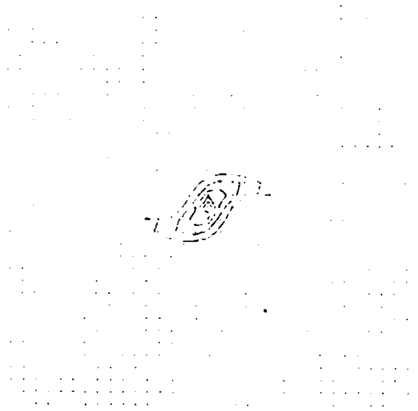


Figure 9. Vorticity Contours after 50 Steps,
2 Core Separation, With Confinement

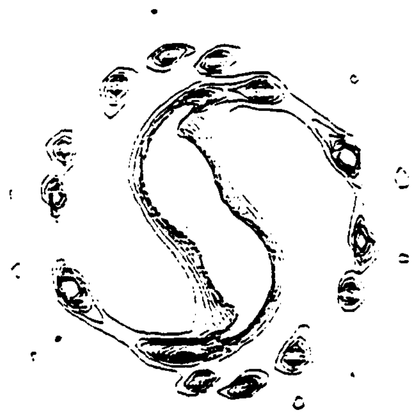


Figure 10. Vorticity Contours

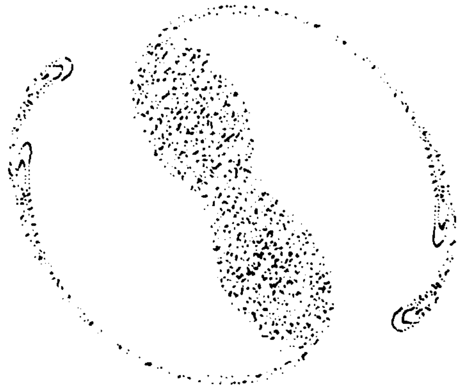


Figure 11. Point Vortices

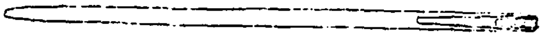


Figure 12. Initial Vorticity Contours

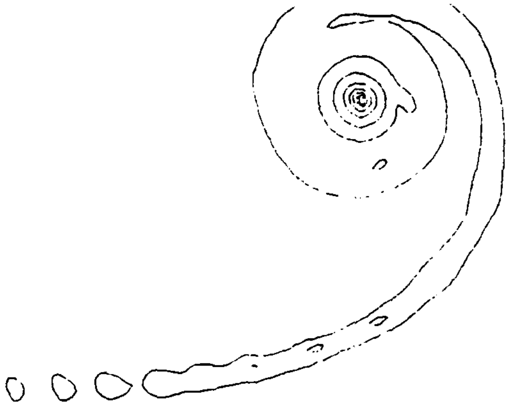


Figure 13. Vorticity Contours after 280 Steps,ε=.0005

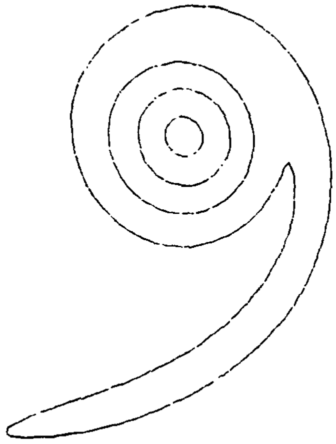


Figure 14. Vorticity Contours after 280 Steps, Without Confinement

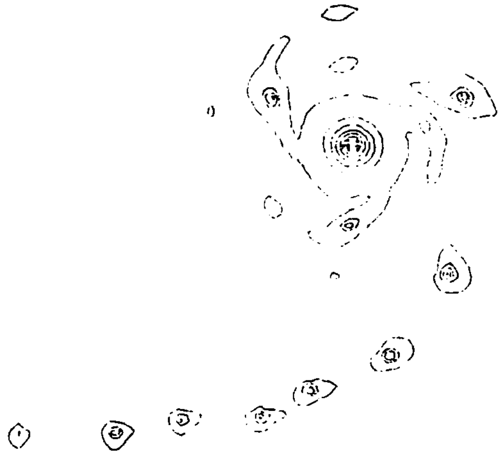


Figure 15. Vorticity Contours after 280 Steps, ε=.002

AGRAWAL, BARNETT, AND ROBINSON: VORTEX BREAKDOWN ON A DELTA WING

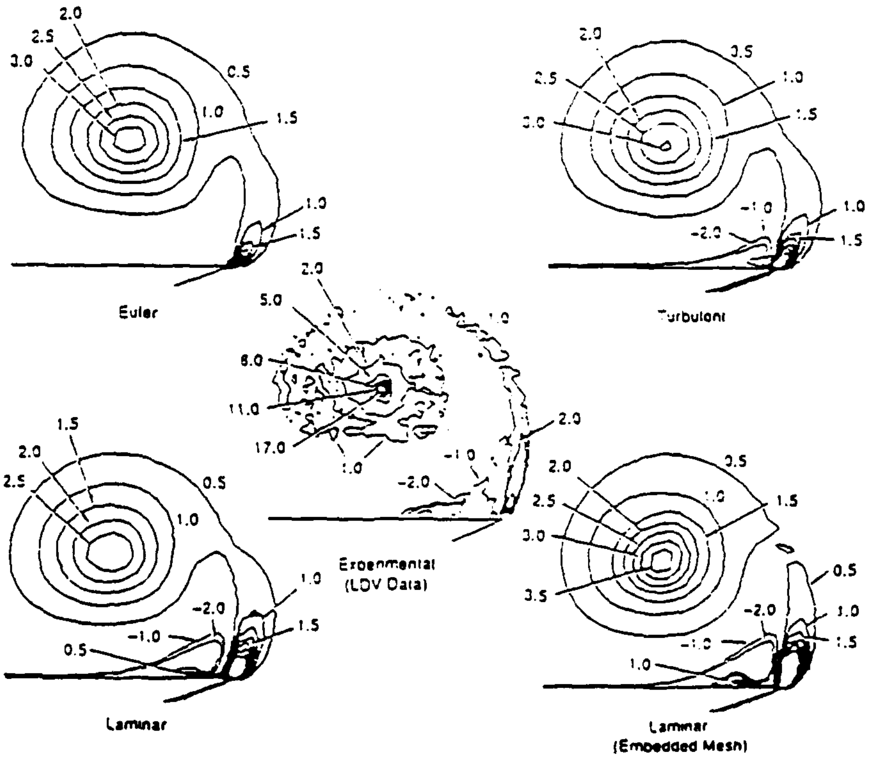


Figure 16. Vorticity Contours on a Delta Wing

# APPENDIX -II

# The Application of Vorticity Embedding to the Computation of Advancing Rotor Flows

**K. Ramachandran**
Flow Analysis, Inc.
Ames Research Center
Moffett Field, California

**S. Schlechtriem**
RWTH University of Aachen
Aachen, Germany

**F.X. Caradonna**
Aeroflightdynamics Directorate
ATCOM
Ames Research Center
Moffett Field, California

**John Steinhoff**
University of Tennessee Space Institute
and
Flow Analysis Inc.,
Tullahoma, Tennessee

## Abstract

A new method has been developed for computing advancing rotor flows. This method uses the Vorticity Embedding technique, which has been developed and validated over the last several years for the hovering rotor problems. In this work, the unsteady full potential equation is solved in an Eulerian grid with an embedded vortical velocity field. This vortical velocity accounts for the influence of the wake. Dynamic grid changes that are required to accommodate prescribed blade motion and deformation are included using a novel grid blending method. Free wake computations have been performed on a two-bladed AH1-G rotor at low advance ratios including blade motion. Computed results are compared with experimental data. The sudden variations in airloads due to blade-vortex interactions on the advancing and retreating sides are well captured though the magnitudes of these changes are under-predicted. Computed wake geometries and their influence on the aerodynamic loads at these advance ratios are also discussed.

## List of Symbols

| | |
|---|---|
| $C_n$ | = normal force coefficient |
| $M$ | = local Mach number for blade section |
| $M_T$ | = hover tip Mach number |
| $r$ | = spanwise distance along the rotor blade |
| $R$ | = radial distance to rotor tip |

*Presented at the American Helicopter Society 49th Annual Forum, St. Louis, Missouri, May 19-21, 1993. Copyright ©1993 by the American Helicopter Society, Inc. All rights reserved.*

| | |
|---|---|
| $\Gamma$ | = rotor bound circulation, first Clebsch variable |
| $\lambda$ | = wake geometry parameter, second Clebsch variable |
| $q_v$ | = vortical velocity components |
| $\mu$ | = rotor advance ratio |
| $\Omega$ | = rotor rotational speed |
| $\chi$ | = Wake coordinates in computational space |
| $\psi$ | = rotor azimuthal angle, deg. |
| $C_t$ | = section thrust |

## Introduction

It has long been recognized that the ability to compute rotor wake formation and convection is the single most important element required for the prediction of rotor performance, vibratory loads, and acoustics. Almost all forward-flight, free-wake prediction methods have been boundary-integral methods. These are typified by the vortex-lattice, lifting-line method used in the well-known CAMRAD/JA [1] and the more recent curved-wake-element, lifting-surface methods developed by Quackenbush et.al. [2]. These are incompressible, inviscid methods, however, and the detailed prediction of rotor loads (in response to the wake-induced inflow) requires auxiliary data or analyses.

In order to obviate much of the present need for airfoil tables, several hybrid schemes have been developed , wherein an inner CFD potential code is coupled either to a comprehensive code (containing a vortex-lattice wake model) [3, 4, 5] or to a free-wake lifting line rotor analysis code [6]. Coupling is achieved by providing a partial inflow from the comprehensive to the CFD code and in turn returning the blade loads to the comprehensive code. The partial inflow includes geometric, blade motion, and wake inflow effects and is imposed through a transpiration condition at the blade surface. For efficiency, the CFD calculations are performed outside the trim loop. The CFD solution for lift is specified as a base solution inside the trim loop. These coupled methods combine an incompressible wake model with a 3-D, unsteady blade solution which treats almost all but the strongest of transonic flows (local Mach numbers less than about 1.3 - a result of the potential approximation). A separate model for the wake effects is required due to the fact that the conventional potential method constrains the wake to lie on a grid plane.

Methods based on Euler or Navier-Stokes equations, though not restricted by a Mach number limits or wake location constraints, are computationally much more demanding, . Recently, Srinivasan et.al. developed a Navier-Stokes method to compute the rotor/wake system [7] in hover. However, structured ENS methods rapidly dissipate vortical structures, such as tip vortices, as a result of numerical diffusion. Such numerical dissipation is minimized by means of high-order schemes and dense grids but the computational cost is considerable. In order to obviate this problem, Wake and Egolf [8] recently coupled a Navier-Stokes flow solver with a lifting line free-wake code for the wake influence. These computations were performed on a massively parallel computer.

At present, there is only one CFD method that can compute rotor/wake flows with no numerical dissipation and with computationally reasonable grid requirements. This is the method of Vorticity Embedding [9] – a combined Eulerian–Lagrangian method. The absence of wake dissipation is due to the fact that the shed circulation is not carried by the grid, but rather by a sheet of convecting wake markers (a Lagrangian tracking process), whose circulation is impressed on the adjacent grid points as a local vortical velocity distribution. These impressed vortical velocity distributions are used as a forcing function for a standard, Eulerian, full-potential flow solver. This approach, implemented in the HELIX-I code, has been successfully applied to the prediction of hover wakes and performance [10].

This paper describes the application of this technique for the computation of low advance ratio forward flight wherein the wake effects are extremely important. This method is efficient in convecting wakes without numerical diffusion and at the same time can treat non-linear unsteady transonic flows that occur near the tip. External coupling is needed only to account for blade motion and deformation. The main objective of this work is to demonstrate that this new CFD based method can successfully model complex advancing rotor wakes and related interactions. A description of the various modules that are required to accomplish this, is provided. The computations shown here represent the first efforts to compute forward flight rotor/wake flows using Vorticity Embedding. They demonstrate the feasibility of a unified rotor-wake CFD computation.

## Solution Procedure

Since the development of the original Vortex Embedding method, a number of modifications and enhancements have been made to accommodate forward flight computations. These modifications have been implemented into a new code HELIX-II. Some important features of HELIX-II are: i) the Unsteady Full Potential Equation is solved and computations are performed along the entire azimuth (360°). ii) Newton-type sub-iterations are performed for better convergence and time accuracy. iii) An improved wake convection procedure computes the Clebsch variables used in the algorithm more efficiently by using computational coordinates. iv) Prescribed blade motions and elastic deformation are included by modifying tne grid near the rotor using a simple blending function method [11].

A schematic of the HELIX-II solution procedure is shown in Figure 1. It can be seen that each time step of the free-wake calculation comprises three major modular computations: i) The Lagrangian wake convection and vortical velocity calculation, ii) incorporating necessary blade motion by modifying the grid and iii) solution of the unsteady mass conservation equation in generalized coordinates. In this section, each module is described.

### Vorticity Embedding

The wake is represented by a set of nodes distributed along the azimuth at constant intervals and along the rotor span (Figure 2) . An undistorted wake is generally used to begin the solution process. The azimuthal node intervals are determined from the time step used for the calculation and radial nodes are situated at grid cell centers. The wake convection procedure for a new time step involves the following steps: i) The inertial coordinates of the blade fixed grid corresponding to the new time step are obtained. ii) A set of new marker nodes is added at the updated blade trailing edge location. iii) All wake nodes are displaced to follow the local flow. This is accomplished using a search algorithm which determines the Eulerian grid cells in which the nodes are located. The three components of velocity that are available at the grid nodes are then interpolated to the marker nodes using trilinear interpolation. Then the wake nodes are moved using:

$$\vec{X}_{l,t+\Delta t} = X_{l-1,t} + V_{intp}\Delta t \qquad (1)$$

Since the azimuthal wake node interval and the time step chosen for the computation are the same, it is possible to associate the node $l$ at the current time to the node $l-1$ at previous time step. In addition, the cell identifying procedure provides the marker computational coordinates. They are used for vortical velocity computation. These integrated marker nodes define the new wake for the present

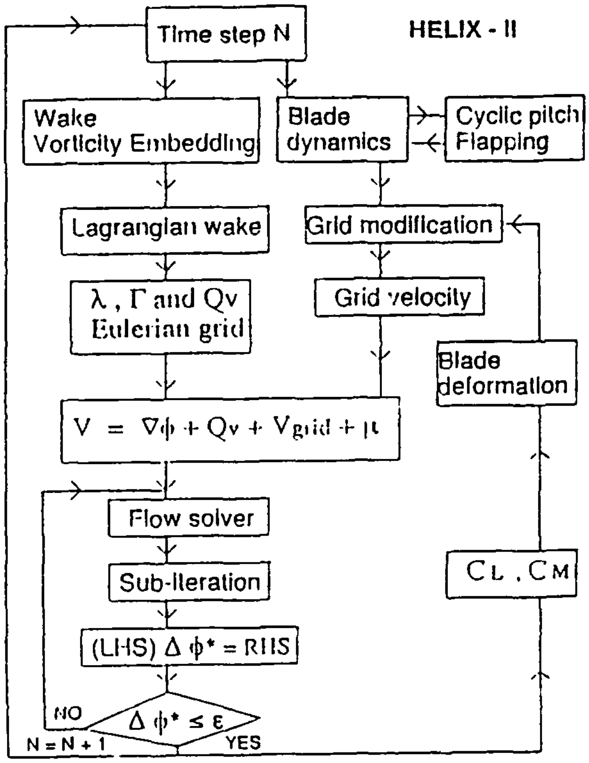


Figure 1: **Schematic of the HELIX-II Solution Procedure**

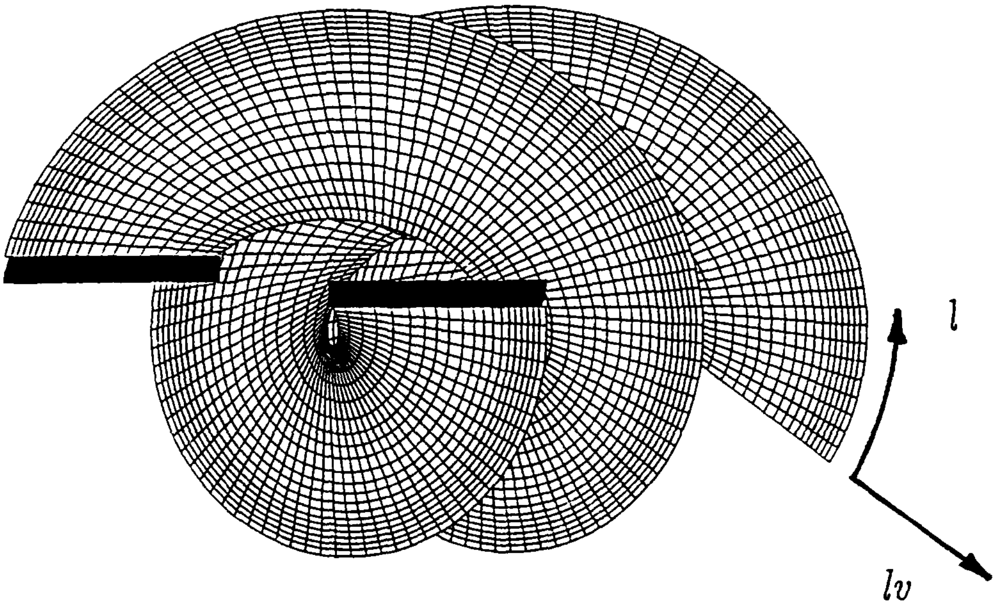


Figure 2: **The Lagrangian wake coordinate system represented by a network of nodes in azimuthal and radial directions**

the present time step. The wake geometry and the blade loading determine the convergence of the solution for that particular time step. In this module, the search algorithm requires the maximum amount of computing time. The present algorithm is very fast and general. An exhaustive search is performed only once in the beginning and subsequent searches use information from earlier computations. A uniform azimuthal node distribution is needed for wake integration. When a large time step (such as 4 degrees) is chosen this may result in poor wake resolution in regions near the blade where the grid is fine. Without enough nodes, the resulting vortical velocity computation will be inaccurate. This is due to the fact that, the vorticity spreading distance is chosen based on the computational cells and at least one wake node is required on every cell in the azimuthal direction. In order to prevent this, a sub-nodal interpolation scheme is used which sub-divides the computed wake between nodes. The wake, with these added sub-nodes mimics the local time stepping approach of the hover code, HELIX-I and is used only for vortical velocity computation. The sub-division is performed as follows: Let $\vec{\chi}_{l,lv}$ be the computed computational coordinate of the wake node $(l, lv)$. Here $l$ is the azimuthal index and $lv$ is the radial index. Define,

$$N_l = \vec{\chi}_{l+1,lv} - \vec{\chi}_{l,lv} \tag{2}$$

If $N_l$ is greater than 1 , $N_l$ sub-nodes are to be added between nodes $l$ and $l+1$. These sub-nodes will have the azimuthal grid indices as their $X$-coordinate and the other two coordinates are obtained using weighted interpolation. That is :

$$\vec{\chi}_{sl+1,lv} = \vec{\chi}_{l+1,lv} F_1/F_3 - \vec{\chi}_{l,lv} F_2/F_3 \tag{3}$$

where$F_1$, $F_2$ and $F_3$ are suitable weighting functions. $\Gamma_{l,lv}$, the wake node circulation is also interpolated in a similar manner to the sub-nodes. This new sub-divided wake with at least one sub-node in each azimuthal cell is used for vortical velocity calculations.

Once the wake is obtained for any given time step, it is embedded into the Eulerian grid as a vortical velocity. That is, the circulation of the convecting Lagrangian wake nodes are impressed on the adjacent grid points as a local vortical velocity distribution. This procedure is described in Ref. [9] and can be briefly stated as: i.) Compute Lagrangian wake. ii.) From each Eulerian grid point compute the distance of a wake panel. If this distance is larger than a specified smearing distance that grid node will not get any contribution from that panel. Otherwise, a shape parameter $\lambda_{i,j,k}$ and a strength parameter $\Gamma_{i,j,k}$ are determined at these grid points. iii.) Using these variables compute $\vec{q_v}$ as:

$$\vec{q_v} = \Gamma_{i,j,k} \vec{\nabla} \lambda_{i,j,k} \qquad (4)$$

This vortical velocity field which is normal to the sheet is solenoidal and accounts for the wake vorticity. In forward flight, the wake is time dependent and there is no single wake geometry. One periodic solution may require several azimuthal revolutions of computations and each revolution in turn has as many wake geometries as the number of time steps. For this reason an efficient procedure is required. A new scheme has been developed which is very fast. In this: i.) The wake convection and the computation of Clebsch constants are done simultaneously. That is, in order to convect the wake, the grid velocities will have to be interpolated to the wake nodes. This requires the identification of Lagrangian nodes in the computational space. They are the integer component of $\vec{X}_{l,lv}$. ii.) Now a vorticity spreading distance $D$ is defined in the computational space which specifies the number of cells over which $\vec{q_v}$ will be spread. The circulation of a given node, $(l, lv)$ is impressed over a region given by,

$$\vec{R_{i,j,k}} = int\left(\vec{X}_{l,lv}\right) \pm D \qquad (5)$$

using Clebsch variables $\Gamma$ and $\lambda$. Since these two variables are scalars they are invariant and can be transported to the rotor grid to obtain $\vec{q}_v$. iii.) A grid node can get a contribution from more than one Lagrangian wake node. Hence these contributions are added. $\Gamma$ and $\lambda$ are computed using interpolation like formulae :

$$\begin{aligned} \Gamma_{i,j,k} &= \frac{\Sigma \Gamma W_{i,j,k}}{\Sigma W_{i,j,k}} \\ \bar{\lambda}_{i,j,k} &= \frac{\Sigma S_n W_{i,j,k}}{\Sigma W_{i,j,k}} \qquad (6) \\ \lambda_{i,j,k} &= \tfrac{1}{2} \sin \tfrac{\pi \bar{\lambda}}{2D} \end{aligned}$$

where $W_{i,j,k}$ is a weighting function given by,

$$W_{i,j,k} = 1 - \frac{\Delta s^2}{D^2} \qquad (7)$$

$\Delta s$ is the distance of a wake node from the grid point and $S_n$ is the signed normal distance of a grid node from a wake panel. These functions are in computational space. Finally, $\vec{q}_v$ is computed using Equation (4) Here, the gradient is computed using the same box scheme as used for the potential calculation.

## Full Potential Formulation

Once $\vec{q}_v$ is obtained the mass conservation equation,

$$\frac{\partial}{\partial \tau} + \vec{\nabla} \cdot \left(\rho \vec{V}\right) = 0 \qquad (8)$$

is solved in the Eulerian grid. Here $\vec{V}$ is given by,

$$\vec{V} = \vec{\nabla}\phi + \vec{q}_v + +\vec{V}_b \qquad (9)$$

where $\vec{V}_b$ is the velocity due to blade motion. Equation (8) is solved using the scheme described in Ref. [9]. The salient features of this solver are: i.) Equation (8) is discretized using finite-volume method. ii.) At each time step, Newton-type sub-iterations are performed for rapid convergence and time accuracy. iii.) At each time step, local time-linearization is performed to provide a good initial guess for the Newton iterations. iv.) Circulation convection is solved in the wake before it is completely replaced by the Lagrangian wake.

Only a brief description of the time-linearization is provided in this section. A more detailed description can be found in Ref. [12]. Eqn. (8) can be represented as

$$F(\phi) = 0 \qquad (10)$$

where $\phi$ is the unknown to be solved for at every grid point at the $n+1$ time level, and $n$ is the temporal index. The Newton iteration for the solution to the above equation is,

$$F(\phi^*) + \left.\frac{\partial F}{\partial \phi}\right|_{\phi=\phi^*} (\phi - \phi^*) = 0 \qquad (11)$$

where $\phi^*$ is the currently available value of $\phi$ at the $n+1$ time level. At convergence,

$$\Delta\phi^* = \phi - \phi^* \qquad (12)$$

will approach zero. Equation (8) can be discretized as

$$\begin{aligned}\frac{\rho^{n+1}-\rho^{n}}{\Delta\tau} &+ \frac{\partial}{\partial\xi}(\rho hU) \\ &+ \frac{\partial}{\partial\eta}(\rho hV) + \frac{\partial}{\partial\zeta}(\rho hW)\Big|^{n+1} = 0\end{aligned} \tag{13}$$

Using expansions similar to that of Equation (11) for $\rho^{n+1}$ and other flux terms we obtain,

$$\begin{aligned}\frac{\rho^{*}-\rho^{n}}{\Delta\tau} + \frac{\partial\rho}{\partial\phi}\Big|_{\phi=\phi^{*}}\Delta\phi^{*} &+ \left[f + \frac{\partial f}{\partial\phi}\Delta\phi^{*}\right]_{\xi} \\ &+ \left[g + \frac{\partial g}{\partial\phi}\Delta\phi^{*}\right]_{\eta} \\ &+ \left[p + \frac{\partial p}{\partial\phi}\Delta\phi^{*}\right]_{\zeta} = 0\end{aligned} \tag{14}$$

In this equation $f, g$ and $p$ are the flux terms and $h$ is the transformation matrix. This equation can be rearranged with $\Delta\phi^{*}$ terms on the left hand side and terms involving $^{*}$ and $^{n}$ levels on the right hand side. The LHS is factored using approximate-factorization [9]. After the rearrangement and factorization the discretized equation looks like

$$L_{\xi}L_{\eta}L_{\zeta}\Delta\phi^{*} = RHS \tag{15}$$

We solve for $\Delta\phi^{*}$ and at the end of the sub-iterations, $\phi^{*} \Rightarrow \phi^{n+1}$. In our scheme the spanwise marching is performed explicitly. Finally, the circulation convection equation is solved in the wake. At the end of the sub-iteration process, the solver provides the total velocity field for wake convection and aerodynamic loads for aeroelastic computation (if a structural module is included).

### Blade Motion

In this section the mechanism to include the prescribed blade motion, such as cyclic pitch and flapping, are described. In addition, the same approach can be used to account for aeroelastic deformation. If $\Theta_0, \Theta_{1c}$ and $\Theta_{1s}$ are the cyclic pitch motion harmonics, for each time step each blade sectional grid is rotated by $\Theta_b$, where

$$\Theta_b = \Theta_0 + \Theta_{1c}\cos(\Psi) + \Theta_{1s}\sin(\Psi) \tag{16}$$

The original grid and the rotated grids are then combined using blending functions. That is

$$\vec{X}_{new} = \vec{X}_{rot}f_{ij} + \vec{X}_{unrot}(1 - f_{ij}). \tag{17}$$

Similarly, the flapping motion is defined by,

$$\beta = \beta_0 + \beta_{1c}\cos(\Psi) + \beta_{1s}\sin(\Psi) \tag{18}$$

The flapping deflection for a given radial station, $r$, is given by

$$\Delta y_{flap} = r\sin(\beta) \tag{19}$$

Using this equation the rotor blade deflection at every span station is computed. It is then used to translate the blade in the normal direction and a similar blending procedure confines the changes to regions near the blade. This process is illustrated in Figure 3. Finally, using the new and old coordinates, the velocity due to blade motion is computed as,

$$\vec{V}_b = \frac{\vec{X}_{t+\Delta t} - \vec{X}_t}{\Delta t} \tag{20}$$

## Results and Discussions

Forward flight computations have been performed on the two-bladed AH1-G rotor at low advance ratios. This rotor, which has a relatively simple geometry, has been chosen for our first validation effort. The open literature contains an exhaustive set of flight test data for a wide range of advance ratios. A rotor-fixed H-H mesh was generated using a two-step algebraic scheme. First, a planar H-H mesh is generated at each radial station with specified outer boundaries. The axial boundaries are chosen to be approximately at 1 radius. Then, the planar grids are transformed to a series of cylindrical grids with constant radii. Once the grid is generated on a single blade it is rotated by an appropriate angle to obtain the grids for the other blades. The AH1-G rotor uses a symmetric airfoil and has an aspect ratio of 9.8. Two different grids with 350,000 and 425,000 points have been used for our investigation. Typically, about 200 points are distributed along the azimuth, 50 in the normal direction, and 40 in the radial direction. A portion of the grid is shown in Figure 4. The twist and collective pitch angle are built into the grid by rotating the section at each radial station. Two different flight conditions have been chosen corresponding to advance ratios of 0.19 and 0.15. The lateral and longitudinal cyclic pitch variations have been obtained from Ref. [13]. Also, the zeroeth (coning)

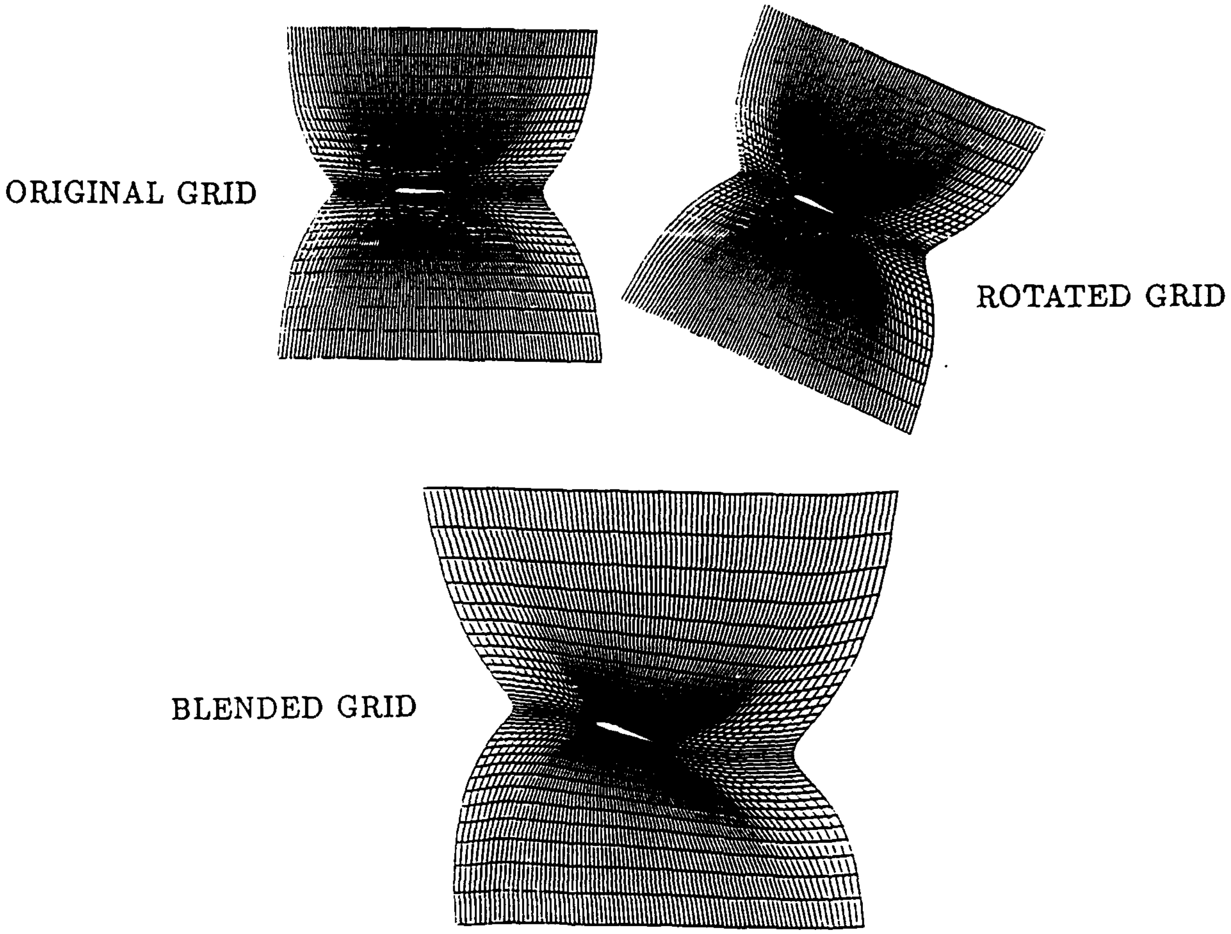


**Figure 3: Blending method for including dynamic grid changes**

and first blade flapping harmonics are input from this report for a particular trim condition. The tip Mach number is 0.65.

The initial undistorted wake is obtained using a uniform axial flow. The undistorted wake has 21 nodes distributed along the spanwise direction and 100 azimuthal nodes. The radial circulation distribution is input from an earlier hover calculation and is constant along the azimuth. The starting wake is very approximate. It is possible to start the solution process with a more accurate wake from an earlier computation. However, in the absence of such information the present starting procedure is used. This results in a longer computing time - requiring about 6 - 8 revolutions before a periodic solution is obtained. Each wake node is displaced using the interpolated velocities obtained from the flow solver. At every time step new nodes are added along the rotor span. In the present calculation, for every set of nodes added a set of far wake nodes is dropped thus maintaining the total number of azimuthal nodes a constant. Thus the free wake computation is performed over 400 degree azimuth and the influence of the far wake included using the wake nodes from earlier revolutions. In Figure 5 the computed wake geometry is plotted at two different azimuths corresponding to 90 and 180 degrees. This is obtained after 6 rotor revolutions and is fully converged. The rolled-up tip vortex regions can clearly be seen. The wake rolls up quickly and convects fairly close to the blade. At these two azimuthal locations, their proximity causes a particularly strong interaction.

Contours of constant circulation in the wake, after 7 revolutions of the blade, are shown in Figure 6. It can be seen here that after about 5 revolutions a periodic state is reached. Also, the progressive development of the blade vortex interaction region over the blade on the retreating side is clearly illustrated. These interactions begin outboard at the second revolution and spreads over entire the blade.

In Figure 7 the azimuthal variation of lift is plotted along the span. The two distinct regions of blade vortex interactions can be seen in this figure. On the advancing side, it happens at about 75 degrees near the tip (outboard) and shifts gradually to about 110 degrees traveling inboard along the span. On the retreating side, similar interactions occur starting at about 275 degrees inboard and slowly moving to about 300 degree and approaching the tip. For this case the periodic solution has been obtained after about 6 rotor revolutions of computations. With a better starting solution this can be reduced to about 3 revolutions.

In Figure 8 the computed normal load variation ($C_N$ ) is plotted for different radial stations as a function of the azimuth. These computations have been performed at an advance ratio of 0.15. The strong blade vortex interactions can be seen on the advancing side at the azimuth of 90 degrees and on the retreating side at about 270 degrees. The computed solutions are compared with those obtained using CAMRAD/JA, a comprehensive code [1]. Blade elastic deformation was not included and thus conditions were identical to those used for HELIX-II runs. For HELIX-II computations, the blade cyclic and flapping harmonics have been obtained from the comprehensive code. A uniformly good comparison is seen, especially in the inboard regions. Maximum deviations are observed at the 90 degree position on the advancing side. On the retreating side, the magnitude of the wake interactions are somewhat under predicted.

In Figure 9 the HELIX-II computed solutions at an advance ratio of 0.19 have been compared to the flight test data obtained from Ref. [13]. The blade-motion coefficients used for this case has been obtained from the flight test data. Good comparisons can be seen inboard. The magnitude of the blade vortex interactions have been somewhat under-predicted near the tip. This is probably due to two reasons. First, the excessive spreading of the vorticity, which is performed over certain number of user specified grid cells. Using a large smearing distance is analogous to having a fat core in conventional lifting-line models. This large smearing affects the accuracy of the solution especially on the advancing side. Generally, a certain minimum amount of wake spreading is required for stability. In order to improve the accuracy the solution can be recomputed using a smaller spreading distance using a converged wake geometry. The wake will not be recomputed but the core will be redefined. In this manner, a new core, almost an order of magnitude smaller, can be obtained without affecting the stability. In addition, the coarseness of the grid would cause excessive smearing. In order to prevent this, using the existing wake, the solution can be recomputed over a short interval using a finer grid. This interval would cover the region of BVI both on the advancing and retreating sides. With a better definition of the wake region interactions would become stronger. The use of a very fine grid for the entire azimuth will be computationally expensive but if it is used only over a short range it should be acceptable.

The choice of time step is another factor determining the accuracy. For the computations shown, a time step of 4 degrees is chosen for free-wake calculation. Again, a smaller time step can be used over a short range and the accuracy can be improved. These possibilities will be explored and future computations will include these features.

## Conclusion

The method of Vorticity Embedding has been used to construct the first self-contained CFD model of a rotor in forward flight. The method (embodied in the code, HELIX-II) predicts the three dimensional, compressible (transonic) flow about a rotor, which both generates and interacts with it's wake system. Although the entire multi-rotor and wake system is discretized, the use of a dissipationless method permits the use of a smaller grid than can be achieved by any structured grid method. The use of a relatively small grid, combined with a potential based method, makes the scheme highly efficient. It has been found the wake solution achieves periodicity (beginning with an undisturbed starting condition) in about 6 rotor revolutions. The method has been used to predict the low advance ratio flow on the AH1-G rotor. Comparisons with flight loads data and CAMRAD/JA computations are highly promising and show that CFD can indeed be used for the treatment of these highly complex flows. Although the present rotor-wake computation is probably the most complex application of Vorticity Embedding to date, the method is intrinsically flexible and amenable to many uses. Vorticity Embedding therefore promises to make a signigicant contribution both to our understanding of vibratory loads, performance, and acoustics and also to the flow predictive methods required by rotorcraft community.

## Acknowledgements

This work was partially supported by Army Research Office under Contract No. DAAL03-89-C-0027. We would like to thank John Bridgeman for many useful suggestions throughout the course of this work. Francisco Hernandez provided CAMRAD/JA and flight test results. Computations were performed using the Numerical Aerodynamic Simulation facility at NASA Ames Research Center.

## References

[1] Johnson, W., "A Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics, Johnson Aeronautics Version," Johnson Aeronautics, 1988.

[2] Bliss,D.B., Teske, M. E., and Quackenbush, T. R., "A New Methodology for Free Wake Analysis Using Curved Vortex Elements," NASA CR 3958, Dec. 1987.

[3] Strawn, R. C., and Caradonna, F. X., "Conservative Full Potential Model for Unsteady Transonic Rotor Flows," *AIAA Journal*, Vol. 25, No. 2, p193, February 1987.

[4] Strawn, R. C., and Tung, C., "Prediction of Unsteady Transonic Rotor Loads With a Full-Potential Rotor Code," presented at the *43rd Annual Forum of the American Helicopter Society*, St. Louis, MO, May 18-20, 1987.

[5] Torok, M. S., and Berezin, C. R., " Aerodynamic and Wake Methodology Evaluation Using Model UH60-A Experimental Data," *presented at the AHS 48th Annual Forum*, Washington D.C., June 3-5, 1992.

[6] Michea, B., Desopper, A. and Costes, M.,"Aerodynamic Rotor Loads Prediction Methods with Free Wake for Low Advance Ratio Descent Flights," presented at the *13th European Rotorcraft Forum*, Arles, France, Sept 11-13 1992.

[7] Srinivasan, G. R., Raghavan, V., and Duque, E.P.N., "Flowfield Analysis of Modern Helicopter Rotors in Hover by Navier-Stokes Method," presented at the *AHS/RAeS International Technical Specialists' Meeting on Rotorcraft Acoustics and Rotor Fluid Dynamics*, Philadelphia, PA, Oct 15-17, 1991.

[8] Wake, B., and Egolf, T. A., "Initial Validation of an unsteady Euler/ Navier-Stokes Flow solver for Helicopter Rotor Airloads in Forward Flight," presented at the *American Helicopter Society International Technical Specialists' Meeting on Rotorcraft Basic Research*, Atlanta, GA, Mar. 1991.

[9] Ramachandran, K., "Free Wake Analysis of Helicopter Rotor Flows in Hover using a Finite Volume Technique,", Ph.D Thesis, University of Tennessee, Knoxville, 1987.

[10] Ramachandran, K.,Tung, C., and Caradonna, F.X., "Rotor Hover Performance Prediction Using a Free Wake, CFD Method," Journal of Aircraft, Vol. 26, No. 12, pp 1105-1110, Dec 1989.

[11] Steinhoff, J. S., " A Blending Method for Grid Generation," Journal of Computational Physics, Vol. 65,p 370,1986.

[12] Shankar, V., Ide, H., Gorski, J., and Osher, S., "A Fast, Time-Accurate, Unsteady Full-Potential Scheme," *AIAA Journal*, Vol. 25, No. 2, Feb. 1987, pp. 230-238.

[13] Gross, J. L., and Watts, M. E., "Tip Aerodynamic and Acoustics Test," NASA Reference Publication 1179, December 1988.

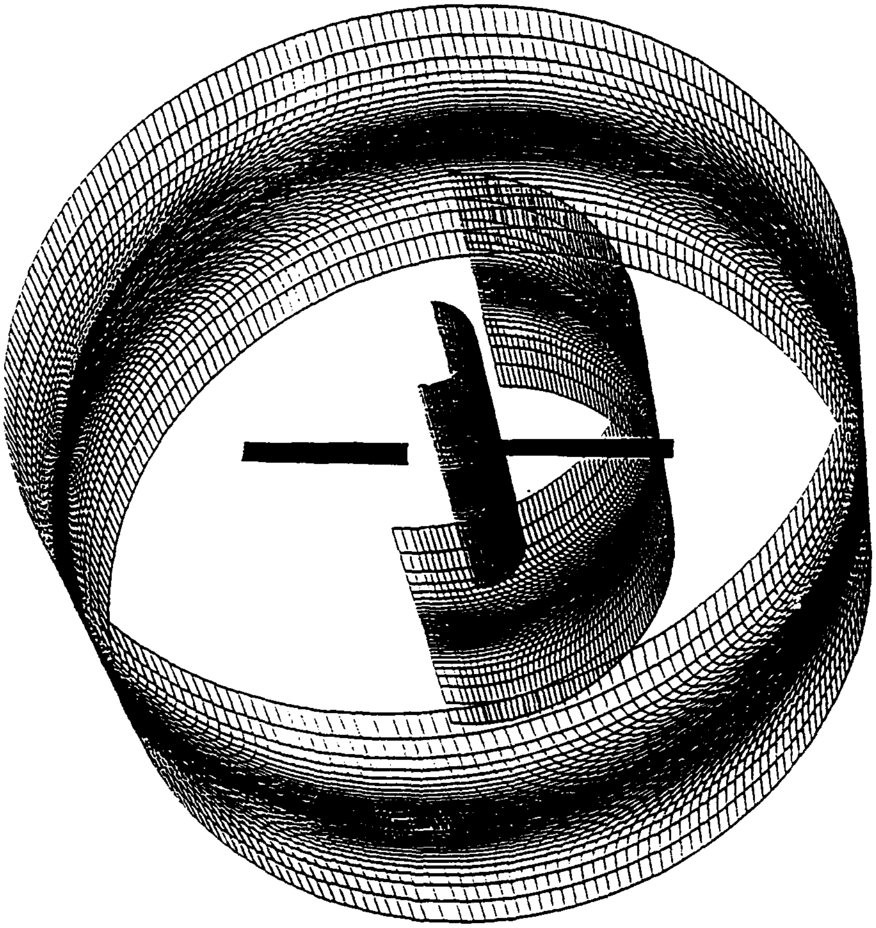


Figure 4: Rotor blade fixed grid used for HELIX-II computations

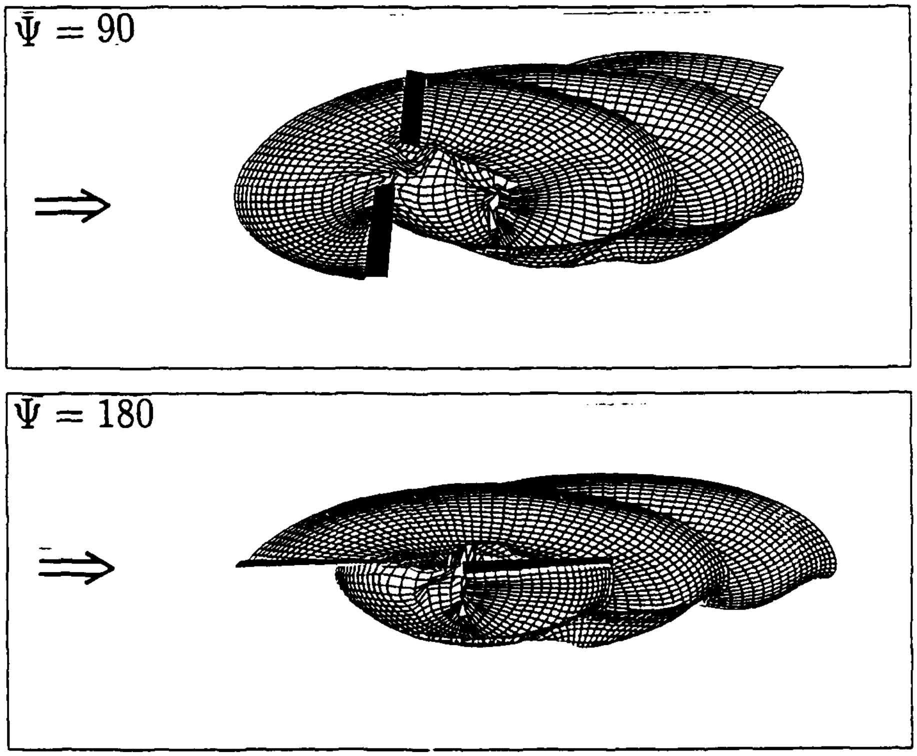


Figure 5: Computed wake geoemtry at 90 and 180 degree azimuths, $\mu = 0.19$

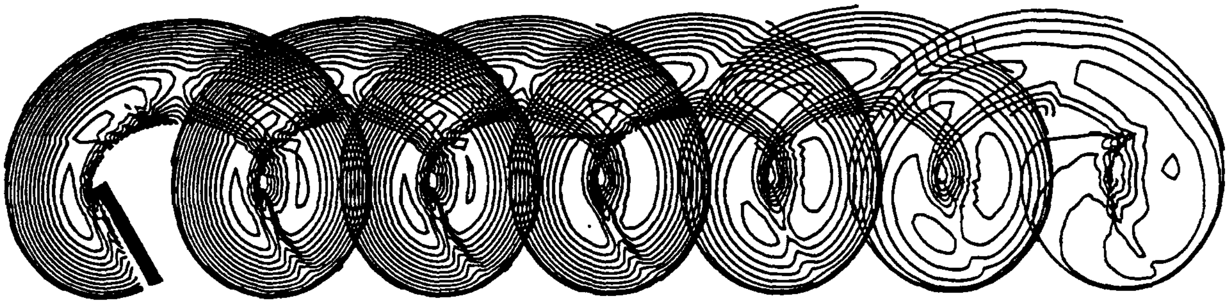


Figure 6: Contours of constant circulation in the wake for 7 rotor revolutions, $\mu = 0.19$

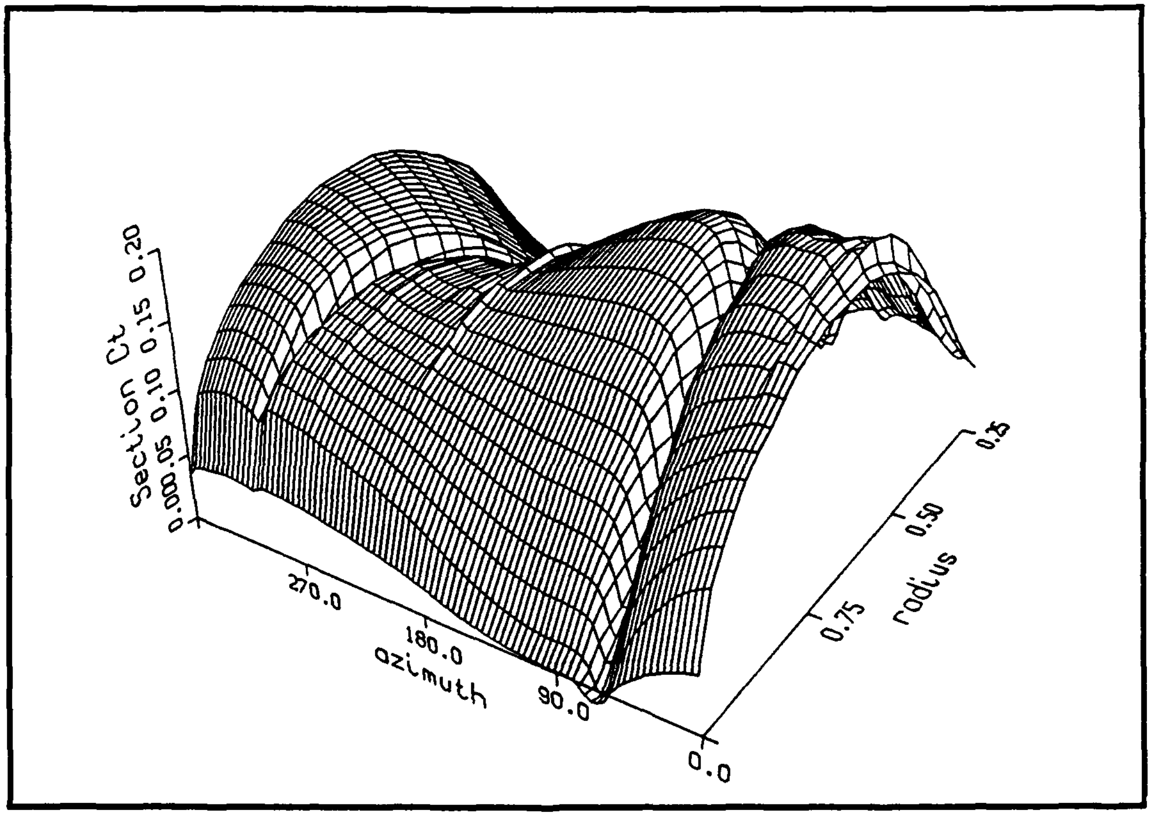


Figure 7: Computed blade airloads for AH1-G as a function of azimuth and rotor radius,$\mu = 0.19$

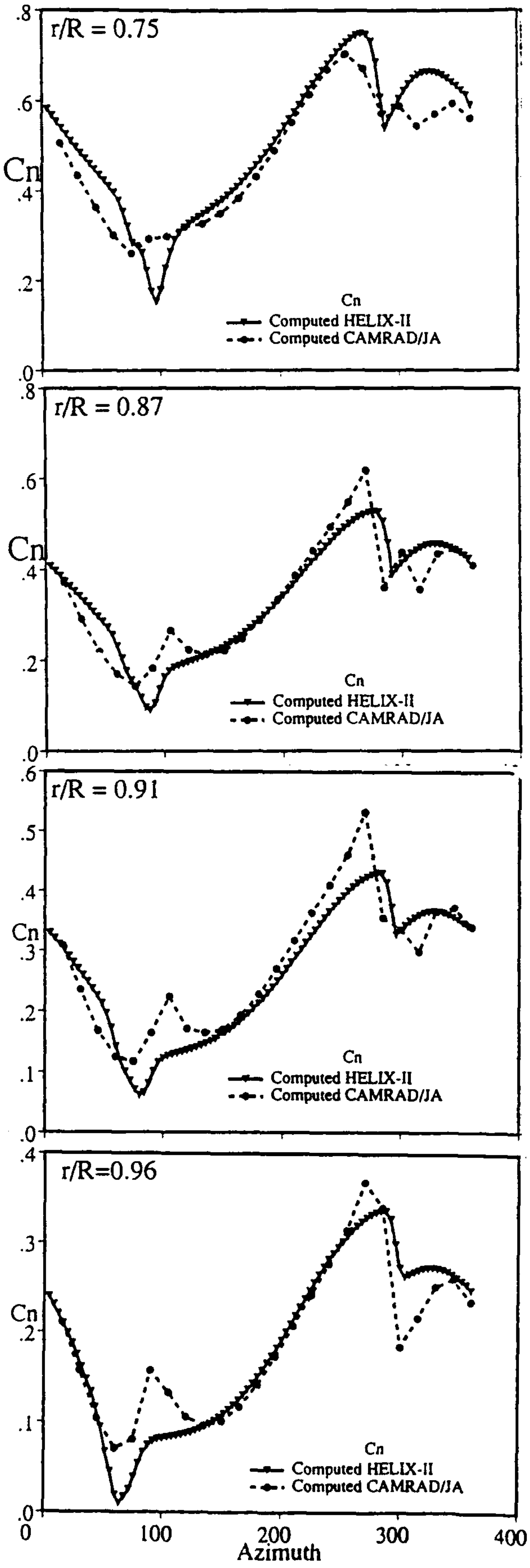


Figure 8: Comparison of airloads with CAMRAD/JA computations, $\mu = 0.15$

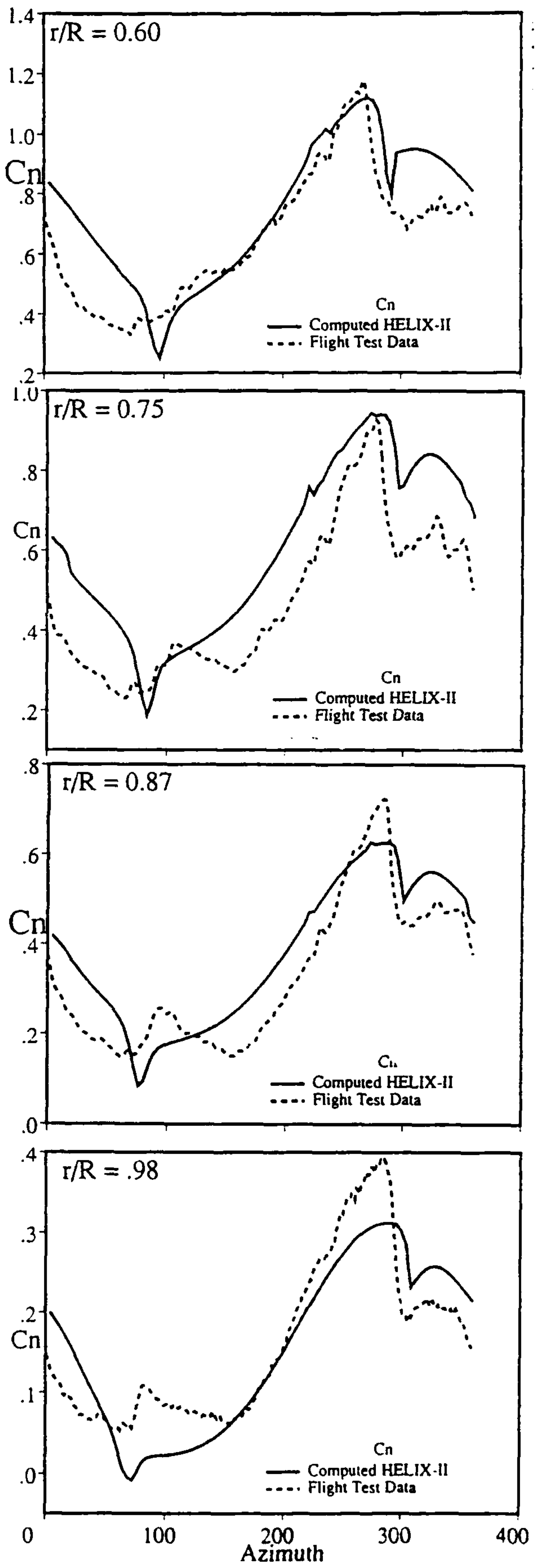


Figure 9: Comparison of computed airloads with flight test data, $\mu = 0.19$